



**Universidade Federal Rural de Pernambuco**  
**Pró-Reitoria de Pesquisa e Pós-Graduação**  
**Programa de Pós-Graduação em Informática Aplicada**

**NIELSON AVELINO DE SANTANA**

**UMA METODOLOGIA PARA AVALIAÇÃO DE DESEMPENHO DE  
APLICAÇÕES EM AMBIENTE DE COMPUTAÇÃO MÓVEL EM  
NUVEM**

**Recife**

**2016**



**Universidade Federal Rural de Pernambuco**

**Pró-Reitoria de Pesquisa e Pós-Graduação**

**Programa de Pós-Graduação em Informática Aplicada**

**UMA METODOLOGIA PARA AVALIAÇÃO DE DESEMPENHO DE  
APLICAÇÕES EM AMBIENTE DE COMPUTAÇÃO MÓVEL EM  
NUVEM**

Dissertação apresentada ao Programa de  
Pós-Graduação em Informática Aplicada  
como exigência parcial à obtenção do  
título de Mestre em Informática Aplicada

**Linha de Pesquisa:**

**Modelagem Matemático Computacional e Simulação**

**Orientador (a): Prof. Dr. Fernando Antonio Aires Lins**

**Co-orientador (a): Prof.(a) Dr.(a) Erica Teixeira Gomes de Sousa**

**Recife**

**2016**

Dados Internacionais de Catalogação na Publicação (CIP)  
Sistema Integrado de Bibliotecas da UFRPE  
Biblioteca Central, Recife-PE, Brasil

S232m Santana, Nielson Avelino de  
Uma metodologia para avaliação de desempenho de aplicações em  
ambiente de computação móvel em nuvem / Nielson Avelino de Santana. –2016.  
64 f. : il.

Orientador: Fernando Antonio Aires Lins.  
Coorientadora: Erica Teixeira Gomes de Sousa.  
Dissertação (Mestrado) – Universidade Federal Rural de Pernambuco,  
Programa de Pós-Graduação em Informática Aplicada, Recife, BR-PE,  
2016.

Inclui referências.

1. Computação móvel 2. Avaliação de desempenho 3. Computação móvel em  
nuvem 4. Simulação I. Lins, Fernando Antonio Aires, orient. II. Sousa, Erica Teixeira  
Gomes de, coorient. III. Título

CDD 004



**Universidade Federal Rural de Pernambuco**  
**Pró-Reitoria de Pesquisa e Pós-Graduação**  
**Programa de Pós-Graduação em Informática Aplicada**

**Uma Metodologia para Avaliação de Desempenho de Aplicações em Ambiente  
de Computação Móvel em Nuvem**

**NIELSON AVELINO DE SANTANA**

Dissertação julgada adequada para  
obtenção do título de Mestre em  
Informática Aplicada, defendida e  
aprovada por unanimidade em 31/08/2016  
pela Banca Examinadora.

Orientador:

---

---

Prof. Dr. Fernando Antonio Aires Lins

Programa de Pós-Graduação em Informática Aplicada - UFRPE

Banca Examinadora:

---

---

Prof. Dr. Gustavo Rau de Almeida Callou

Membro Interno – Programa de Pós-Graduação em Informática Aplicada (PPGIA) -  
UFRPE

---

---

Prof. Dr. Gilberto Amado de Azevedo Cysneiros Filho

Membro Interno – Programa de Pós-Graduação em Informática Aplicada (PPGIA) -  
UFRPE

---

---

Prof(a). Dr(a). Juliana Basto Diniz

Membro Externo – Programa de Pós-Graduação em Tecnologia e Gestão em  
Educação a Distância (PPGTEG) – UFRPE

*À minha mãe, à minha família e  
a todos que acreditaram em  
mim.*

## AGRADECIMENTOS

À Deus.

À **toda minha família** pelo apoio e incentivo.

À meu orientador, **Fernando Antonio Aires Lins**, por sua excelente orientação, paciência, conselhos e sempre ter dado o incentivo necessário para a realização do trabalho.

À minha co-orientadora, **Erica Teixeira Gomes de Sousa**, por sua excelente orientação, paciência, disposição, assistência e apoio.

A todos os Professores/Educadores que contribuíram para minha formação, em especial **Paulo Renato Firmino, Rodrigo Souza, Emerson Trevisan, Sueli Trevisan**.

A todos que contribuíram direta ou indiretamente em minha vida.

À **FACEPE** pelo apoio financeiro.

À **Universidade Federal Rural de Pernambuco** pela infraestrutura para a realização do trabalho.

*“Um passo à frente e você não está no mesmo lugar.”*

**Chico Science**



## RESUMO

Um dos principais obstáculos para o desenvolvimento de aplicações móveis mais elaboradas e com características cognitivas, isto é, aplicações que usem tecnologias como visão computacional, reconhecimento de voz vem a ser a baixa autonomia energética dos dispositivos móveis. A partir deste cenário de dificuldade no desenvolvimento destas aplicações, surge a computação móvel em nuvem. A computação móvel em nuvem é um paradigma para aplicações móveis que visa estender tanto o poder de processamento dos dispositivos móveis quanto aumentar a autonomia da bateria através da transferência das tarefas de processamento para a nuvem. Neste contexto, a proposta deste trabalho é uma metodologia integrada para avaliação de desempenho de aplicações em ambiente de nuvem móvel. A metodologia é composta de duas fases. A primeira fase apresenta a técnica de avaliação de desempenho por medição, e nela são apresentadas as atividades relativas ao entendimento do ambiente, como por exemplo, coleta das métricas e apresentação dos resultados. A segunda fase por sua vez, apresenta a técnica de avaliação de desempenho por meio da simulação. Em virtude da metodologia ser integrada, os dados coletados na primeira fase (fase de medição) são usados para o desenvolvimento de modelos de simulação. A metodologia proposta neste trabalho é aplicada em um estudo de caso que consiste na execução do *benchmark Linpack* no próprio dispositivo móvel e na nuvem computacional, e este estudo objetiva avaliar se a nuvem pode realmente executar tarefas com menor tempo de resposta que o dispositivo móvel. O estudo de caso também proporcionou a verificação da aplicabilidade da metodologia em ambientes de computação móvel em nuvem. Como resultado do estudo de caso, há indícios que a execução de atividades na nuvem computacional, pode ser mais rápida que a execução local em certos contextos, entretanto é necessária a realização de novos estudos com outros *benchmarks* além dos matemáticos. Por fim, a metodologia se mostrou adequada à realização do estudo de caso.

**Palavras-chave:** computação móvel; avaliação de desempenho; computação móvel em nuvem, simulação.

## ABSTRACT

One of the main drawbacks to the development of mobile applications with compute intensive capabilities, that is, applications based on technologies such as natural language processing, voice recognition and augmented reality, is the low energy-autonomy of cell phones. In the context of this restricted scenario set in the development of these applications, it rises the mobile cloud computing. The mobile cloud computing is a paradigm for mobile applications that extends the processing efficiency as either the battery energy life by leveraging the demanding tasks to the cloud. To this end, the proposal of this work is an integrated methodology for performance evaluation of applications in the mobile cloud environment. The proposed methodology is presented in two phases. The first phase presents the performance evaluation technique by measurement. It details the steps which includes the context problem comprehension, metrics collection tasks and the results demonstration. The second phase, in turn, presents the performance evaluation technique by simulation. Since the approaches are linked, the data collected in the first phase (measurement step) will be used as input for the development of the simulation models. Furthermore, the given methodology is applied in a case study, which consisted at the execution of the Linpack benchmark in the device and cloud scenarios, in order to assess whether the cloud can be more performatic over the local execution. The case study was considered also to check if the proposed methodology was adequate in test of a real world experiment. The experiment results showed that the mobile cloud computing execution can be more efficient than the local processing in the mobile environment. Although the promising results, it is still required further studies with other benchmark models besides the mathematical ones approached in this work. Finally, we conclude that the proposed methodology presented is adequate for the case study experiments.

**Keywords:** cloud computing; performance evaluation techniques; mobile cloud computing; simulations.

# SUMÁRIO

<b>SUMÁRIO</b> .....	vi
Lista de Figuras.....	ix
Lista de Tabelas.....	ix
Lista de Acrônimos.....	x
<b>1. Introdução</b> .....	<b>11</b>
1.1. Motivação e justificativa.....	3
1.2. Objetivos.....	4
1.3. Contribuições.....	4
1.4. Estrutura da dissertação.....	5
<b>2. Fundamentação teórica</b> .....	<b>6</b>
2.1. Fundamentos de computação móvel em nuvem.....	7
2.1.1. Computação móvel.....	7
2.1.2. Computação em Nuvem.....	8
2.1.3. Computação móvel em nuvem.....	10
2.2. Avaliação de desempenho.....	11
2.2.1. Técnicas de avaliação de desempenho.....	12
2.3. ARENA.....	15
2.3.1. Elementos de um modelo de simulação No ARENA.....	16
2.3.2. Input Analyser.....	18
2.3.3. Process Analyser.....	18
2.4. Considerações finais.....	19

3. Trabalhos relacionados.....	19
3.1. Medição.....	20
3.2. Simulação.....	22
3.3. Avaliação comparativa.....	22
3.4. Considerações finais.....	24
4. Metodologia de avaliação de desempenho de Aplicações em ambientes de computação móvel em nuvem.....	24
4.1. Fase de Medição.....	27
4.1.1. Definição de Objetivos e sistema.....	27
4.1.2. Seleção de métricas.....	28
4.1.3. Definição de parâmetros que afetam o desempenho do sistema.....	30
4.1.4. Definição de período de coleta.....	30
4.1.5. Definição de tipo e níveis de carga de trabalho.....	31
4.1.6. Execução da medição e obtenção dos resultados.....	33
4.1.7. Análise estatística dos resultados.....	37
4.2. Fase de Simulação.....	38
4.2.1. Desenvolvimento do modelo de simulação.....	38
4.2.2. Refinamento do Modelo.....	39
4.2.3. Verificação e Validação do modelo.....	39
4.2.4. Análise de novos cenários.....	40
4.3. Modelo de Simulação.....	41
4.4. Considerações finais.....	41
5. Estudo de caso.....	42
5.1. Estudo de caso 1: <i>Linpack</i> .....	44

5.1.1. Fase de medição.....	45
5.1.2. Fase de Simulação.....	50
5.2. Considerações finais.....	56
6. Conclusões e trabalhos futuros.....	56
6.1. Conclusões.....	57
6.2. Contribuições.....	57
6.3. Limitações.....	59
6.4. Trabalhos Futuros.....	59
7. Referências.....	60

## Lista de Figuras

Figura 2.1 Técnicas de avaliação de desempenho.....	13
Figura 2.2 Modelo de Simulação no Arena para execução de tarefas no dispositivo móvel.....	17
Figura 4.1 Metodologia de avaliação de desempenho de aplicativos em ambiente de computação móvel em nuvem.....	26
Figura 4.2 Ferramenta de avaliação de desempenho proposta.....	36
Figura 4.3 Modelo de simulação para avaliação de desempenho no dispositivo móvel e na nuvem computacional.....	41
Figura 5.1 Visão geral do ambiente de experimentos.....	43
Figura 5.2 Comparação tempo de resposta no local vs nuvem.....	49
Figura 5.3 Comparação tempo de resposta no local vs nuvem.....	50
Figura 5.4 Comparação do tempo médio de resposta: sistema real vs duplicação de recursos.....	55
Figura 5.5 Comparação do tempo médio de resposta: sistema real vs duplicação de recursos.....	56

## Lista de Tabelas

Tabela 2.1 Comparação de abordagens para a avaliação de desempenho.....	15
Tabela 2.2 Módulos básicos do Flowchart do Arena.....	16
Tabela 2.3 Modulos de dados do Arena.....	17
Tabela 3.1 Sumário dos trabalhos relacionados.....	23
Tabela 4.1 Combinações possíveis para dois fatores usando <i>Full Factorial</i> .....	33
Tabela 4.2 Comparação de ferramentas de coleta de métricas no sistema <i>Android</i> .....	37
Tabela 5.1 Exemplo de coleta de métricas de tempo de resposta e tempo de execução.....	46
Tabela 5.2 Carga de trabalho.....	48
Tabela 5.3 Sumarização do percentual de outliers.....	49
Tabela 5.4 Cenários usados para a geração da distribuição de probabilidade.....	51
Tabela 5.5 Distribuições de probabilidade para o <i>Linpack</i> .....	51
Tabela 5.6 Validação do modelo de simulação baseado no tempo de resposta medido no sistema real em relação à simulação para o ambiente local.....	53
Tabela 5.7 Validação do modelo de simulação baseado no tempo de resposta medido no sistema real em relação à simulação para o ambiente de nuvem.....	54

## Lista de Acrônimos

DOE: Design of Experiments.....	32
IaaS: <i>Infrastructure as a service</i> .....	10
MCC: <i>Mobile Cloud Computing</i> .....	5
PaaS: <i>Platform as a service</i> .....	10
SaaS: <i>Software as a service</i> .....	10
SDK: <i>Software Development Kit</i> .....	10
SLA: <i>Service-Level Agreement</i> .....	25



# 1. INTRODUÇÃO

Atualmente, pode-se afirmar que a sociedade vive na era da computação móvel (BAHL et al., 2012)(KUMAR; LU, 2010). Este fato pode ser observado ao considerar o número expressivo de usuários de tecnologias móveis em específico *smartphones*. A estimativa de pessoas que usaram ao menos uma vez por mês o *smartphone* em 2017 é de 2,28 bilhões (IDC, 2015). Outra estimativa importante para o ano de 2017 é o que o gasto no mercado de anúncios para dispositivos móveis será acima de \$100 bilhões (EMARKETER, 2015). Os estudos também apresentam tendências de aumento nos próximos anos para o número de usuários quanto o valor gasto com anúncios na plataforma móvel. Tais evidências apontam que os *smartphones* são tão importantes para a sociedade quanto os computadores *desktops/notebooks*.

Dada à importância que a computação móvel vem recebendo e apesar dos esforços realizados pela indústria, diversos são os desafios existentes para o desenvolvimento de aplicações mais elaboradas, isto é, aplicações que segundo (CHEN et al., 2015), possam ter características cognitivas através do uso tecnologias como inteligência artificial, sinterização de texto em áudio, reconhecimento de voz, processamento de linguagem natural. O uso de tais tecnologias constitui um desafio para os dispositivos móveis principalmente pela necessidade de recursos computacionais (CPU, Memória, IO) para por exemplo, realizar o reconhecimento de voz ou realizar algum aprendizado de máquina. Além da demanda por recursos computacionais, tais aplicações também demandam alto consumo da bateria do dispositivo móvel, o qual já é um desafio em virtude da baixa autonomia energética dos dispositivos móveis (MIETTINEN; NURMINEN, 2010)(KUMAR; LU, 2010)(BAHL et al., 2012).

Em paralelo está o surgimento, desenvolvimento e popularização da computação em nuvem. A computação em nuvem é um modelo que permite acesso ubíquo, conveniente e sob demanda a um conjunto compartilhado de recursos computacionais (redes, servidores, armazenamento, aplicações e serviços) que tem

rápida provisão e liberação com o mínimo de esforço de gerenciamento (MELL; GRANCE, 2011).

Nesse contexto, surgiu o paradigma de computação móvel em nuvem (*mobile cloud computing*), que vem a ser a integração da computação móvel com a computação em nuvem e visa estender a capacidade computacional dos dispositivos móveis por meio do provisionamento de recursos computacionais da nuvem (KUMAR; LU, 2010). Tal objetivo seria alcançado por meio da migração e execução das tarefas do dispositivo móvel para a nuvem, técnica essa chamada de *offloading* (BAHL et al., 2012)(KUMAR; LU, 2010). Em outras palavras, a computação em nuvem oferece a possibilidade de economia de energia bem como melhoramento no desempenho em virtude da execução de tarefas intensivas computacionalmente sejam realizadas fora do dispositivo móvel (MIETTINEN; NURMINEN, 2010)(DINH et al., 2013).

Baseado nas questões apresentadas verifica-se a necessidade de estudos relacionados à avaliação de desempenho na área de computação móvel em nuvem, tendo em vista a considerável quantidade e diversidade de aplicações móveis existentes. É fato que alguns tipos de aplicações podem apresentar desempenho expressivo ao usar nuvem enquanto outras podem não apresentar ganho significativo. Neste contexto, a avaliação de desempenho possibilita o estudo de inúmeros fatores que estão envolvidos na proposta de computação móvel em nuvem, tais como a latência, tempo de resposta e capacidade de processamento do servidor e do dispositivo móvel.

Para alcançar este objetivo, o trabalho proposto apresenta uma metodologia para a avaliação de desempenho de aplicativos móveis em ambiente de computação móvel em nuvem, baseado especificamente na plataforma *Android* (GOOGLE, 2016a). Esta metodologia é composta por um conjunto de atividades que estão agrupadas em duas fases: medição e simulação. Através da execução destas atividades, o usuário poderá fazer um estudo de desempenho, desde o seu projeto até execução e análise, das aplicações no ambiente de nuvem móvel, possibilitando a geração de informações relevantes para possíveis tomadas de decisão (ex.: usar

ou não a nuvem e, qual a configuração de máquina virtual mais adequada para usar a nuvem).

## **1.1. MOTIVAÇÃO E JUSTIFICATIVA**

A computação móvel é usada atualmente por cerca de 2 bilhões de usuários (EMARKETER, 2014). Embora o número de usuários que requerem cada vez mais recursos e autonomia de bateria seja expressivo, a natureza móvel e o tamanho reduzido dos dispositivos móveis fazem com que recursos como o poder de processamento, memória RAM e armazenamento secundário sejam escassos (QI; GANI, 2012). Mesmo nos casos em que os dispositivos móveis se equiparam aos computadores domésticos, emerge o desafio da baixa autonomia energética dos dispositivos móveis, que pode inviabilizar o desenvolvimento de aplicações mais elaboradas como as feitas para computadores (MIETTINEN; NURMINEN, 2010)(KUMAR; LU, 2010)(BAHL et al., 2012).

Em virtude da escassez de tais recursos (em principal bateria), o desenvolvimento de aplicações mais rebuscadas se torna um desafio para esta plataforma móvel. Exemplos de aplicações mais rebuscadas são aplicações interativas e com o uso de tecnologias como reconhecimento de voz, linguagem natural, visão computacional, realidade aumentada entre outras tecnologias que poderiam facilitar mais a vida em vários aspectos (SATYANARAYANAN, 1996) (BAHL et al., 2012).

Neste contexto, a partir do uso de tais tecnologias a computação móvel em nuvem possibilitaria a criação de aplicações mais interativas e com características cognitivas que poderiam auxiliar pessoas com necessidades especiais como por exemplo, um deficiente visual (CHEN et al., 2015) (SATYANARAYANAN et al., 2014). Contudo, um desafio que aplicações móveis precisam ultrapassar para serem consideradas efetivamente interativas vem a ser o tempo de resposta ao usuário. O tempo de resposta - que é composto tanto da latência da rede quanto do processamento - deve ser baixo o suficiente para que não comprometa a experiência do usuário (SATYANARAYANAN et al., 2014).

Desenvolver aplicações móveis que tenham tempo de resposta baixo o suficiente para que seja imperceptível na interação humano computador representa um grande desafio tendo em vista tanto a distância física da nuvem (*Data Center*) até o usuário além do tempo de processamento a ser realizado. Desta forma, o estudo de avaliação de desempenho se mostra imprescindível, pois poderá ajudar a encontrar soluções para a otimização do sistema.

## **1.2. OBJETIVOS**

O trabalho tem como principal objetivo a proposição de uma metodologia de avaliação de desempenho de aplicações em ambiente de computação móvel em nuvem. A metodologia consiste da combinação do método de avaliação por medição e simulação e será detalhado em atividades, seguindo um fluxo de passo a passo.

Os objetivos secundários da proposta são:

- Desenvolvimento e validação de modelo de simulação para a avaliação de desempenho de ambiente de computação móvel em nuvem;
- Estudo de impacto do *offloading* de tarefas do dispositivo móvel para a nuvem em relação ao tempo de resposta a partir do estudo de caso;
- Desenvolvimento de uma ferramenta para a geração de carga de trabalho (no dispositivo móvel e na nuvem) e também para a coleta de métricas como tempo de execução, tempo de resposta, tamanho da fila de tarefas.

## **1.3. CONTRIBUIÇÕES**

A principal contribuição desta dissertação é a proposta de uma metodologia integrada para avaliação de desempenho de aplicações em ambiente de nuvem móvel, composta pelas técnicas de avaliação de desempenho baseadas em medição simulação. Essa contribuição é relevante, pois, considerando o estado da arte atual da área, propõe de forma inovadora o uso conjunto de recursos de medição e simulação para o estudo de desempenho.

As demais contribuições do trabalho são:

- O modelo de simulação que permite o estudo de novos cenários e com diferentes cargas de trabalho;
- O estudo de impacto do *offloading* de tarefas do dispositivo móvel para a nuvem, que apresenta o quão eficiente a nuvem pode ser na realização de tarefas computacionalmente intensivas em ambientes *mobile cloud*;
- Uma ferramenta adaptada de (HAMREN, 2015) para a medição foi proposta tendo em vista que as ferramentas existentes não supriam os requisitos para a realização do experimento de avaliação de desempenho. A ferramenta adaptada é composta de dois módulos: um módulo para instalação no servidor e outro para instalação no dispositivo móvel na plataforma *Android*. O módulo do servidor processa e retorna o resultado da computação enquanto o módulo do dispositivo móvel pode executar as tarefas tanto localmente quanto enviar para o servidor na nuvem. A importância da ferramenta se dá ao permitir a realização de experimentos em um ambiente real (local ou nuvem) e além de gerar um arquivo de *log* da coleta de métricas de desempenho para análises subsequentes.

#### **1.4. ESTRUTURA DA DISSERTAÇÃO**

O capítulo 2 apresenta as fundamentações teóricas que serão a base do presente trabalho. Na primeira seção são apresentados os conceitos básicos de *mobile cloud computing (MCC)* ou computação móvel em nuvem. A segunda seção apresenta os fundamentos básicos relacionados à avaliação de desempenho e as técnicas existentes para a realização de avaliação.

O capítulo 3, por sua vez, apresenta uma visão geral da área por meio dos trabalhos relacionados. Para uma melhor visualização, esses trabalhos foram categorizados em: trabalhos que abordam a técnica de medição e trabalhos que abordam a técnica de simulação; após esta apresentação, será apresentada uma análise comparativa a respeito dos trabalhos.

Já o capítulo 4 apresenta a metodologia de avaliação de desempenho que é composta pelas técnicas de medição e simulação. Nele, cada uma das atividades

que compõe a metodologia será apresentada e detalhado. Por fim, é apresentado o modelo de simulação desenvolvido.

O capítulo 5 apresenta estudos de caso de avaliação de desempenho de aplicações no contexto de computação móvel em nuvem e permitirá verificar se a metodologia proposta é adequada. O *benchmark* usado no estudo de caso foi o *Linpack*.

Por fim, o capítulo 6 apresenta as conclusões, as contribuições desenvolvidas no trabalho, as limitações encontradas no trabalho e os trabalhos futuros vislumbrados.

## 2. FUNDAMENTAÇÃO TEÓRICA

O capítulo apresenta os conceitos básicos necessários para o entendimento do trabalho e foi dividido em duas seções. Na primeira serão apresentados os conceitos básicos de *mobile cloud computing (MCC)*. A segunda seção apresenta os fundamentos básicos relacionados a avaliação de desempenho e as técnicas existentes para a realização desta avaliação.

### 2.1. FUNDAMENTOS DE COMPUTAÇÃO MÓVEL EM NUVEM

*Mobile cloud computing (MCC)* é uma área de estudo composta por duas outras áreas de relevância: *mobile computing* e *cloud computing*. Desta forma, antes de apresentar o tema principal, serão apresentados os conceitos básicos das duas áreas que a compõe.

#### 2.1.1. COMPUTAÇÃO MÓVEL

A computação móvel está presente na vida do homem, desde períodos remotos. Uma das primeiras ferramentas de computação móvel foi o ábaco (calculadora) e desta época em diante, conforme o progresso da tecnologia, outras calculadoras e outros dispositivos mais modernos foram desenvolvidos. Atualmente existe uma infinidade de dispositivos de computação móvel que podem ser usados pelo homem, inclusive alguns que podem ser vestidos, como é o caso dos relógios inteligentes.

De forma geral, entende-se por computação móvel sistemas computacionais que mesmo ao serem movidos de um lugar físico para outro, permitem que suas funcionalidades possam ser usadas (B'FAR, 2004). Como exemplo de computação móvel pode-se citar notebooks, calculadoras, celulares, *smartphones*, e *mp3 players*. Contudo, para os objetivos do presente trabalho, o dispositivo também precisa ter conectividade, isto é, receber e enviar dados na *Internet* por meios de conexão sem fio (ARCHER et al., 2012). Ou seja, o presente trabalho se restringirá aos dispositivos com conectividade móvel sem fio (3G, 4G, *WiFi*), em específico os *smartphones*, devido suas limitações de processamento, armazenamento e baixa autonomia energética.

Dispositivos móveis apresentam uma série de restrições como o tamanho, o peso, dentre outras que por consequência implicam no desempenho em termos computacionais dos dispositivos. Mesmo os celulares de última geração, com maior poder de processamento, esbarram no problema da autonomia energética, pois são grandezas inversamente proporcionais, isto é, quanto maior o poder de processamento, maior será o consumo de energia. Entretanto, esse vem a ser um dos requisitos dos usuários, que querem celulares mais robustos e com tela maiores. Neste contexto, algumas das principais limitações inerentes aos dispositivos móveis são (SATYANARAYANAN, 1996):

- **Dispositivos móveis possuem recursos limitados em relação a máquinas estáticas:** apesar do desenvolvimento dos dispositivos móveis, em específico dos *smartphones*, a demanda dos usuários por dispositivos com telas maiores, maior poder de processamento e menor peso, implica diretamente na autonomia energética do dispositivo.
- **Mobilidade é inerentemente perigosa:** a natureza móvel dos dispositivos implica em questões de segurança em relação aos dados que estão armazenados no próprio dispositivo. Exemplos de dados são fotos, dados bancários e agenda telefônica.
- **A conectividade móvel é variável no desempenho e na confiabilidade:** a conexão dos dispositivos móveis possui alta variabilidade, uma vez que os usuários podem usar conexões *WiFi* de alta velocidade ou redes 3G/4G com velocidades bem mais limitada, dependendo da localização do usuário.
- **Autonomia energética limitada:** a baixa autonomia energética é um dos principais problemas que afeta diretamente os dispositivos móveis, em especial *smartphones*, uma vez que esses dispositivos são usados constantemente ao longo do dia pelos usuários, tanto para comunicação quanto para entretenimento.

### 2.1.2. COMPUTAÇÃO EM NUVEM

Computação em nuvem é um modelo computacional no qual recursos como processador, rede, armazenamento e *software* são disponibilizados através da *Internet* e podem ser acessados remotamente (ARMBRUST et al., 2010). A



computação em nuvem oferece algumas vantagens, como permitir aos usuários utilizar recursos elasticamente e de forma *on-demand* além de ter baixo custo (DINH et al., 2013).

Segundo o NIST, uma nuvem deve satisfazer um conjunto de requisitos para ser considerada efetivamente um serviço de nuvem. As características são: *on-demand self-service* – os clientes usarão o serviço de acordo com suas demandas; *broad network access* - cliente poderá acessar de quaisquer meios, tais como *smartphones, tablets*; *resource pooling* – os recursos computacionais são atribuídos aos clientes de forma dinâmica e o cliente não tem controle do recurso que irá usar, *rapid elasticity* – os recursos computacionais se adequam a demanda do cliente; *measured service* – o cliente é cobrado pelo uso (MELL; GRANCE, 2011).

A computação em nuvem pode ser classificada em dois diferentes aspectos (ZHANG; CHENG; BOUTABA, 2010). O primeiro aspecto está associado com quem usa o serviço e o segundo aspecto está relacionado com quem mantém a infraestrutura. No primeiro, a nuvem pode ser classificada em:

- **Nuvem pública:** é uma nuvem fornecida por uma organização externa a uma empresa que contrata o serviço. É o conceito mais próximo ao de computação utilitária, ou seja, ao pagamento de um serviço fornecido sob demanda, como eletricidade ou água.
- **Nuvem privada:** são implementadas e utilizadas internamente em uma organização. Um ponto fraco desse modelo é o fato de que toda a infraestrutura deve ser mensurada e mantida pelo próprio utilizador, o que contraria alguns princípios da computação em nuvem, como a rápida escalabilidade de recursos com custo proporcional à demanda solicitada.
- **Nuvem comunitária:** são nuvens computacionais utilizadas e mantidas internamente por um conjunto de empresas ou instituições, facilitando o compartilhamento de recursos entre as entidades.
- **Nuvem híbrida:** utiliza conceitos de nuvens públicas ou privadas. Este último caso aplica-se, em geral, às nuvens nas quais estão presentes dados ou aplicações estratégicas.

O segundo aspecto no qual a nuvem computacional pode ser classificada é de acordo com o seu modelo de negócio.

- **Infraestrutura como serviço (IaaS):** oferecem ao cliente o provisionamento sob demanda de recursos computacionais fundamentais como máquinas virtuais (VM), armazenamento no qual o cliente poderá executar qualquer software que queira. Como exemplo de IaaS, pode-se citar a Amazon AWS (AMAZON, 2016) e Microsoft Azure (MICROSOFT, 2016).
- **Plataforma como serviço (PaaS):** oferecem como serviço uma plataforma na qual uma camada abstrai os recursos da infraestrutura (tais como máquinas virtuais, sistemas operacionais, armazenamento). Em alguns casos, Paas podem disponibilizar aos clientes um *software development kit* (SDK). Exemplos de PaaS são *Google App Engine* (GOOGLE, 2016b) e *Heroku* (SALESFORCE, 2016).
- **Software como serviço (SaaS):** por sua vez disponibiliza sob demanda aos clientes aplicações como serviço através da *Internet*. Exemplos de provedores de SaaS são *SalesForce* (SALESFORCE, 2016) e *Rackspace* (RACKSPACE, 2016).

Embora o presente trabalho apresente os três tipos de modelo de negócio, o modelo no qual o trabalho se baseou foi o *IaaS*, a partir da nuvem computação da Amazon. Ressalta-se que a escolha de uma *PaaS* ou *SaaS* poderá impactar no desempenho em comparação com a *IaaS*.

### 2.1.3. COMPUTAÇÃO MÓVEL EM NUVEM

Diversas são as definições que podem ser atribuídas a *mobile cloud computing* (MCC). A definição que será adotada no trabalho considera a possibilidade da economia de energia através do *offloading* de tarefas para a nuvem (KUMAR; LU, 2010).

Kumar (KUMAR; LU, 2010) apresenta a computação móvel em nuvem como um novo paradigma para aplicações móveis onde dados e processamento são movidos para computadores com alto poder de processamento que ficam

localizados em nuvens centralizadas. Estas aplicações são acessadas por meio de conexões sem fio (*WiFi/3G*) nos dispositivos móveis.

De forma geral, a computação móvel em nuvem foi desenvolvida em cima de dois pilares: computação móvel e computação em nuvem. Atualmente, com o crescimento do mercado de dispositivos móveis, os usuários estão cada vez mais demandando por novas funcionalidades, maior poder de processamento. Contudo, fazer com que os *smartphones* usem processadores mais poderosos e tenham mais memória implica diretamente no consumo de bateria. Kumar pontua que alguns usuários preferem a maior autonomia energética da bateria ao invés de novos recursos nos *smartphones* (KUMAR; LU, 2010). Neste contexto, considera-se que devido às limitações dos dispositivos móveis, a computação em nuvem poderia estender recursos como a bateria, ou mesmo melhorar o desempenho por meio da migração de tarefas computacionais para serem executadas na nuvem (SATYANARAYANAN, 1996).

De forma geral, o principal objetivo da computação móvel em nuvem é estender recursos do dispositivo móvel bem como melhorar seu desempenho. Em complemento, com o desenvolvimento da área de MCC, será possível criar aplicações mais interativas e com o uso de tecnologias como reconhecimento de voz, linguagem natural, visão computacional entre outras tecnologias que requerem alto poder de processamento (SATYANARAYANAN, 1996).

## **2.2. AVALIAÇÃO DE DESEMPENHO**

Na área de ciência da computação, a busca pelo desempenho foi um dos principais fatores a impulsionar o progresso tecnológico. Esta busca se deu por diversos fatores, dentre eles o econômico, no qual a intenção era obter o maior desempenho com o menor custo. De forma geral, a avaliação de desempenho consiste de um conjunto de técnicas e regras que visam a sistematização no processo de mensuração do desempenho do sistema sendo ele real ou abstrato (JAIN, 1991) (LILJA, 2000).

Contudo, para se realizar a avaliação de desempenho, é necessário o entendimento do sistema a ser avaliado bem como o conhecimento das técnicas de

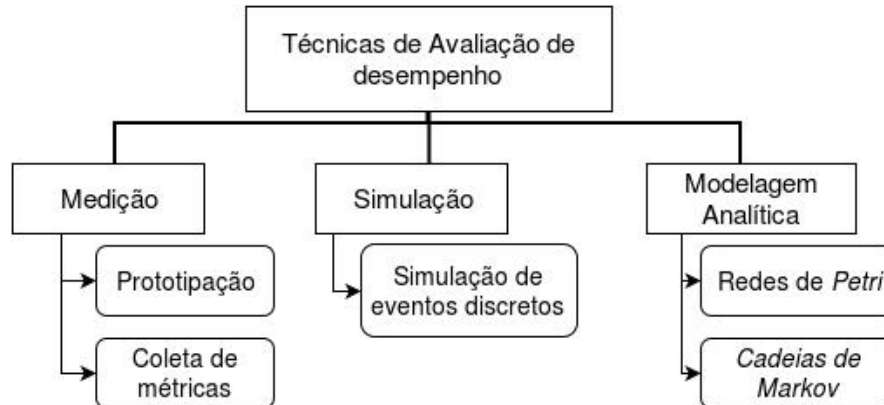
avaliação de desempenho existentes. A análise de desempenho pode ser feita devido a diversos objetivos, dentre os quais:

- **Comparar alternativas** – A comparação de alternativas consiste em escolher entre as alternativas, o que mais se adéqua com a requisitos estabelecidos. Em termos de *hardware*, exemplos de requisitos podem ser: um servidor rápido para operações de escrita e leitura em disco; rápida em termos de processamento.
- **Identificar gargalos e/ou de desperdício de sistemas** – Neste caso, sistemas normalmente possuem gargalos e na medida em que o gargalo é identificado, sua solução reflete em um melhor desempenho. Exemplos de sistemas que possuem gargalos são as aplicações Web. Quando o banco de dados de tais aplicações passa a demorar a responder as buscas, o sistema possui um gargalo com o banco de dados.
- **Determinar configuração ótima dos parâmetros de um sistema** – Determinar a configuração ótima de um sistema permite maximizar o desempenho do sistema sem desperdiçar recursos. Por exemplo, ao se instalar um banco de dados em um servidor, é necessário verificar se as configurações de *hardware* (CPU, memória RAM) serão suficientes para a carga de trabalho prevista.
- **Fazer previsão de desempenho em função do crescimento da carga** – O estudo do desempenho do sistema em função da carga de trabalho é usado em sistemas que podem sofrer alta demanda em pouco tempo. Um exemplo são aplicações web, que podem sofrer picos de usuários e ficar fora do ar.

### **2.2.1. TÉCNICAS DE AVALIAÇÃO DE DESEMPENHO**

De forma geral, uma técnica de avaliação de desempenho consiste de um conjunto de atividades, regras que visam sistematização do processo. A avaliação de desempenho pode ser dividida em três técnicas: medição, simulação e modelagem analítica (JAIN, 1991)(LILJA, 2000). A Figura 2.1 apresenta uma visão geral sobre a área de avaliação de desempenho.

**Figura 2.1 Técnicas de avaliação de desempenho**



A técnica de medição consiste em submeter o sistema a uma determinada carga de trabalho, coletar as saídas geradas (por meio de métricas) e realizar a análise dos resultados coletados. Esta técnica pode ser usada nos casos em que o sistema já exista e embora não ofereça tanta flexibilidade, apresenta resultados acurados. A técnica medição é considerada uma abordagem muito importante no contexto da busca por sistemas com maior desempenho e menor custo (LILJA, 2000). As principais técnicas de medição existentes são:

- **Prototipação** – A técnica de prototipação pode ser aplicada em contextos onde não se possui um sistema real. A prototipação é uma versão simplificada do sistema real e contém as principais características para a análise do sistema. Uma das vantagens da prototipação é a produção de resultados com boa precisão mesmo em estágio inicial do projeto. Já como desvantagens, há o custo da construção do protótipo e pouca flexibilidade para realizar modificações.
- **Monitor ou Coleta de dados** – A técnica de avaliação por meio de coleta de dados permite estudar sistemas reais por meio da coleta das características mais relevantes do sistema. A partir dos dados coletados pelo monitor será possível realizar estudos estatísticos.

A técnica de simulação consiste no uso de programas de modelagem para a criação de modelos de simulação que descrevam os recursos mais importantes do sistema. Após o desenvolvimento do modelo de simulação é preciso verificar e

validar o modelo para assegurar que os resultados são representativos em relação ao sistema real.

As simulações podem ser por eventos discretos ou contínuos. A simulação por eventos discretos usa números inteiros para representar os estados do sistema, e é chamada de simulação de eventos discretos. Esta técnica de simulação é muito usada na área de computação em virtude de ser possível descrever o sistema em termos de números inteiros. Como exemplo, para se descrever um modelo de simulação de um sistema web, algumas variáveis como número de requisições, o número de servidores são números inteiros. A simulação por eventos contínuos por outro lado, usa valores reais para a representação os estados da simulação. Um exemplo de aplicação é na área de química, onde as reações químicas representam um novo estado a cada fração de tempo (JAIN, 1991).

A técnica de modelagem analítica consiste no desenvolvimento modelos e equações matemáticas que descrevam o comportamento do sistema. A técnica de modelagem analítica pode ser usada em contextos em que o sistema real não exista. Embora a técnica seja flexível, os resultados gerados não são tão acurados quanto as outras técnicas. Alguns dos formalismos matemáticos que podem ser usados na modelagem analítica são: cadeias de *Markov* e Redes de *Petri* (BOLCH et al., 2006) (BERTHOMIEU; DIAZ, 1991).

Em se tratando da escolha das técnicas de avaliação, ela pode realizada baseada em alguns fatores como fatores como o estágio do sistema. Por exemplo, técnica de medição só poderá ser aplicada se o sistema existir. Já as técnicas de simulação e de modelagem analítica podem ser aplicadas em qualquer fase do projeto. A Tabela 2.1 apresenta uma compilação dos critérios relacionados a escolha de cada técnica de avaliação (JAIN, 1991) (LILJA, 2000). A partir da tabela, será possível definir qual a técnica de avaliação de desempenho mais adequada, tendo em vista critérios como tempo requerido, flexibilidade, acurácia e custos.

**Tabela 2.1 Comparação de abordagens para a avaliação de desempenho**

<b>Critério</b>	<b>Técnicas de avaliação de desempenho</b>		
	<b>Modelagem analítica</b>	<b>Simulação</b>	<b>Medição</b>
Fase	Qualquer	Qualquer	Prototipação / sistema real
Flexibilidade	Alto	Alto	Baixo
Tempo requerido	Baixo	Médio	Variável
Ferramentas	Analises	Ferramentas de simulação	Instrumentos de medição
Acurácia	Baixo	Moderado	Variável
Avaliação de <i>Trade-off</i>	Fácil	Moderado	Difícil
Custos	Baixo	Médio	Alto
Negociável	Baixo	Médio	Alto

Fonte: Adaptação de Jain (JAIN, 1991) e Lilja (LILJA, 2000)

### A

Tabela 2.1 descreve as características de três técnicas de avaliação distintas (Modelagem analítica, Simulação e Medição). A partir dela, é possível identificar qual a técnica de avaliação de desempenho mais adequada para um determinado contexto. A técnica de medição, por exemplo, pode ser usada em casos onde não se tenha dados sobre o processo e demanda mais tempo para ser desenvolvida, além de ser menos flexível. Por outro lado, se já houver dados disponível sobre o processo, é possível usar a simulação.

## 2.3. ARENA

O Arena é uma ferramenta de simulação de eventos discretos que permite o desenvolvimento de modelos de simulação usando um formalismo gráfico. Um modelo de simulação consiste na aplicação de técnicas matemáticas em sistemas computacionais, permitindo representar uma infinidade de tipos de operações ou processos do mundo real (KELTON; SADOWSKI; SADOWSKI, 2002) (“Arena Simulation Software,” 2016). Como exemplo de processos, pode-se citar o processo de carga e descarga de navios, trens.

A seguir, na seção 2.3.1, será apresentado o formalismo e alguns elementos do modelo de simulação do Arena. Além disso, o Arena disponibiliza algumas ferramentas auxiliares tanto para o desenvolvimento do modelo quanto para a









execução de simulações. As ferramentas a serem apresentadas são o *Input Analyzer* o *Process Analyzer*.

### 2.3.1. ELEMENTOS DE UM MODELO DE SIMULAÇÃO NO ARENA

No Arena, os modelos de simulação são basicamente construídos a partir de blocos (objetos). Os blocos por sua vez, podem ser desenvolvidos a partir de dois módulos. O módulo de diagrama de fluxo (flowchart) e o módulo de dados.

O módulo de fluxo descreve o comportamento das entidades que são usadas no modelo de simulação. Alguns dos módulos de fluxo (flowchart) que são encontrados no módulo básico do Arena são apresentados na Tabela 2.2.

**Tabela 2.2 Módulos básicos do Flowchart do Arena**

<i>Figura</i>	<i>Nome</i>	<i>Descrição</i>
	Create	Inicializa entidades na simulação.
	Dispose	Remove entidades processadas da simulação.
	Process	Processa uma entidade usando um ou mais recursos e requer algum tempo para completar o processo.
	Decide	Decide se uma entidade irá seguir uma ramificação ou outra.
	Batch	Agrupa um determinado número de entidades antes de continuar o fluxo.
	Separate	Duplica entidades para o processamento paralelo.
	Assign	Altera o valor de algum parâmetro (durante a simulação). Ex. tipo da entidade, valor da variável.
	Record	Coleta estatísticas como o número de entidades processadas.



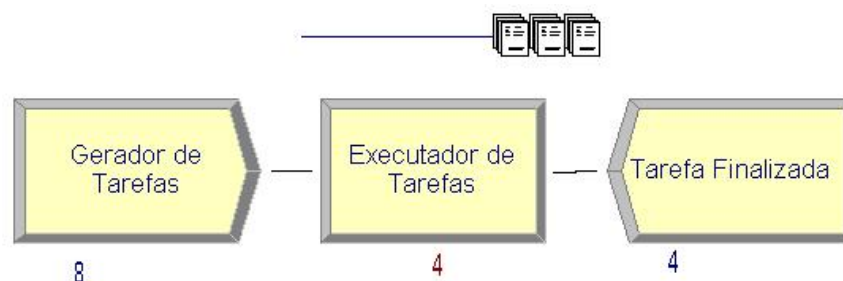
O módulo de dados por outro lado, é usado para definir os parâmetros associados com elementos do modelo tais como *resources*, *entities*, *queue*. A Tabela 2.3 apresenta os principais elementos do módulo de dados.

**Tabela 2.3 Modulos de dados do Arena**

<b>Nome</b>	<b>Descrição</b>
<i>Entity</i>	Define vários tipos de entidades bem como seu estado inicial e que será usado na simulação.
<i>Resource</i>	Define os recursos do sistema incluindo custo e disponibilidade de recursos.
<i>Queue</i>	Define as filas do sistema
<i>Schedule</i>	Com o módulo <i>Create</i> , define o agendamento da criação de novas entidades na simulação.
<i>Variable</i>	Define as variáveis que serão usadas
<i>Attributes</i>	Define os atributos das entidades como por exemplo: tipo, dimensão e valor inicial.

Baseado nos elementos apresentados, a Figura 2.2 é mostra um exemplo de modelo de simulação usando a ferramenta Arena que representa a execução de tarefas do dispositivo móvel na nuvem.

**Figura 2.2 Modelo de Simulação no Arena para execução de tarefas no dispositivo móvel**



No modelo, cada elemento (objeto) possui um propósito:

- **Gerador de tarefas** - define o tipo de entidade e a regra de chegadas de entidades na simulação por meio do agendamento;
- **Executador de Tarefa** - representa o tempo de processamento que a atividade demandará para ser realizada e pode ser expresso por uma distribuição de probabilidade ou por um tempo constante;
- **Fila** - Acima da atividade "Executador de Tarefas" é apresentado uma fila de entidades (objetos) que estão aguardando serem processadas;
- **Tarefa Finalizada** - representa o fim do processamento de uma entidade e a liberação de recursos.

### 2.3.2. INPUT ANALYSER

O Input Analyzer, é uma ferramenta auxiliar do Arena que permite a geração de distribuições de probabilidade a partir de dados coletados. Distribuição de probabilidade é uma ferramenta estatística que permite descrever o comportamento que uma variável poderá assumir ao longo de um espaço de valores.

O funcionamento do Input Analyzer consiste em receber como entrada um conjunto de dados números que descrevem algum comportamento no mundo real. A partir dos dados, o Input Analyzer traça padrões e mapeiam os dados para distribuições de probabilidades como: *Normal, Geométrica, Beta, Exponencial*.

Em adicional, o Input Analyzer usa testes estatísticos como o Chi Square e o Kolmogorov-Smirnov para verificar se as distribuições geradas são representativas em relação aos dados usados. Outra informação importante apresentada pelo Input Analyzer é o erro quadrado. O erro quadrado permite identificar a distribuição de probabilidade que mais se aproximou da distribuição real dos dados coletados. De forma geral, quanto menor o erro quadrado, mais a distribuição de probabilidade se aproximará da distribuição dos dados coletados.

### **2.3.3. PROCESS ANALYSER**

O Process Analyzer é uma ferramenta que auxilia na automatização da realização de experimentos. A partir dele, é possível atribuir parâmetros dinâmicos ao modelo, como o número de replicações, ou alguma variável definida no modelo, como por exemplo, o tempo de início. Permitindo assim que se crie vários cenários e que os mesmos sejam avaliados de uma única vez.

## **2.4. CONSIDERAÇÕES FINAIS**

Este capítulo apresentou os fundamentos teóricos relacionados ao trabalho e consistiu em apresentar as áreas de computação móvel em nuvem e a área de avaliação de desempenho bem como mostrou uma introdução a ferramenta de simulação Arena. No capítulo seguinte serão apresentados os trabalhos relacionados ao trabalho proposto.

### 3. TRABALHOS RELACIONADOS

O presente capítulo apresenta trabalhos relacionados ao tema de avaliação de desempenho em ambiente de computação móvel em nuvem que usem a técnica de avaliação de desempenho por medição ou simulação e que foram aplicados na plataforma *Android*. Até o presente momento, não foram encontrados trabalhos que apresentem metodologia usando as técnicas de medição e simulação para a avaliação de desempenho de ambiente de computação móvel em nuvem, em específico a capacidade de processamento de um dispositivo móvel em relação a nuvem. Algumas das fontes de busca dos trabalhos relacionados foram: *Google Acadêmico*, *Portal de Periódicos da CAPES*.

Outro fator importante na busca dos trabalhos foi a busca por trabalhos recentes que tenham sido publicados com no mínimo 6 anos atrás, de forma a ter um conjunto de trabalhos relativamente recentes em relação ao estado da arte atual.

No intuito de melhor apresentar os trabalhos relacionados, serão usadas três seções para a categorização: avaliação de desempenho pela técnica de medição e avaliação de desempenho pela técnica de simulação e avaliação pela técnica de medição e simulação.

#### 3.1. MEDIÇÃO

Nesta seção serão apresentados os principais trabalhos relacionados a avaliação de desempenho de ambiente de computação móvel em nuvem por meio de medição.

Nesse sentido, (HAMRÉN, 2012) propôs um estudo quantitativo baseado na pergunta: a execução da computação em nuvem pode realmente ser mais rápida que a computação local? O estudo consistiu no desenvolvimento de um aplicativo na plataforma *Android* para avaliar o desempenho de tarefas computacionais sendo executadas na nuvem e no dispositivo móvel. Para alcançar este objetivo, foi implementado um conjunto de *benchmarks* no aplicativo: ordenação de texto, contagem de números primos de 0 a X. Em seguida, o aplicativo foi disponibilizado ao público para ser executado em *smatphones*. Após o termino da execução dos

*benchmarks*, o aplicativo envia os resultados para um servidor para serem analisados. Como principal diferença em relação a metodologia empregada por HAMRÉN e comparação com a apresentada no presente trabalho, HAMRÉN empregou de uma abordagem quantitativa e os experimentos foram realizados por outras pessoas em seus *smarthphones* e em ambientes não controlados. Em contra partida, a metodologia do presente trabalho se concentrou na aplicação do experimento em ambientes controlado, no intuito de ter resultados mais precisos, o que é de grande relevância em um estudo de avaliação de desempenho.

Outro trabalho importante é o proposto por (SILVA; MACIEL; MATOS, 2015) e consiste em um *framework SmartRank*, que possui duas funções: realizar o particionamento da tarefa e o *offloading* de tarefas do dispositivo móvel para a nuvem. Para mostrar a eficiência do *framework*, o trabalho propôs um estudo de avaliação de desempenho em termos de resposta para tarefas de reconhecimento facial de imagens em ambiente de computação móvel em nuvem. Como resultado, o trabalho apresentou que o uso do *SmartRank* pode diminuir em 48% o tempo de resposta para o reconhecimento de faces em fotos ao usar a computação em nuvem. Como destaque, o trabalho detalha o funcionamento e implementação do *benchmark* de reconhecimento de faces em imagens quanto a implementação do *SmartRank*.

Seguindo essa linha, (HAMAD; SAAD; ABED, 2010) apresenta um estudo baseado na comparação dos protocolos REST e SOAP por meio das métricas (tempo de resposta da requisição e no tamanho da mensagem enviada) tendo em vista a escassez de recursos dos dispositivos móveis. O trabalho apresenta as bases teóricas sobre os protocolos SOAP e o REST, explica a implementação do experimento e bem como as configurações do servidor e do dispositivo móvel. O experimento foi composto por dois benchmarks: um para concatenação de *array* de *string* e outro de soma de números decimais de um *array*. Os dois benchmarks foram desenvolvidos usando tanto o protocolo SOAP quanto o REST. Como pontos positivos, o trabalho apresenta em detalhes a metodologia usada para a realização do experimento. A partir dos resultados do experimento, verificou-se que tanto o

tempo de requisição quanto o tamanho das mensagens do protocolo REST era menor em relação ao protocolo SOAP.

### **3.2. SIMULAÇÃO**

Abaixo são descritos os trabalhos que apresentam a proposta de avaliação de desempenho por meio da técnica de simulação. BAHWAIRETH (BAHWAIRETH et al., 2016) apresenta um estudo experimental baseado na comparação de *frameworks* para modelagem e simulação de infraestruturas para o ambiente de computação móvel em nuvem. No estudo, os experimentos de *benchmarks* foram modelados em cada *framework* (*CloudSim*, *CloudAnalyst*, *CloudReports*, *CloudExp*, *GreenCloud*, e *iCanCloud*) (BAHWAIRETH et al., 2016). O trabalho apresentou uma importante contribuição ao ter compilado ferramentas de simulação existentes para a simulação de computação móvel em nuvem. Como desvantagem, os experimentos realizados nos *frameworks* de avaliação de desempenho não foram padronizados, isto é, em cada *framework* foi aplicado um benchmark diferente.

### **3.3. AVALIAÇÃO COMPARATIVA**

Como apresentado nas seções anteriores, existe um número limitado de trabalhos que envolvem as áreas de avaliação de desempenho por meio das técnicas de medição e simulação. De forma geral, a baixa quantidade de trabalhos deve-se ao fato de a área de MCC ser recente e pelo fato do escopo deste trabalho estar direcionado para a avaliação de desempenho (em termos de tempo de execução) do dispositivo móvel em detrimento a outros aspectos, como consumo de energia.

A partir da Tabela 3.1, percebe-se a escassez de trabalhos que envolvam a combinação das técnicas de medição e simulação e sua aplicação em um ambiente real de computação móvel em nuvem. Segundo a tabela, a maioria dos trabalhos ((HAMAD; SAAD; ABED, 2010), (HAMRÉN, 2012), (SILVA; MACIEL; MATOS, 2015)) usaram alguma técnica de medição para realizar o experimento. Por outro lado, poucos são os trabalhos que abordam a simulação em ambiente de computação móvel em nuvem, e cita-se neste caso o trabalho de (BAHWAIRETH et al., 2016).

A Tabela 3.1 apresenta a sumarização baseado nas características dos trabalhos relacionados em relação ao trabalho proposto nesta dissertação. A partir da Tabela 3.1, que se percebe a escassez de trabalhos que envolvam a combinação das técnicas de medição e simulação e sua aplicação em um ambiente real de computação móvel em nuvem. Segundo a tabela, a maioria dos trabalhos ((HAMAD; SAAD; ABED, 2010), (HAMRÉN, 2012), (SILVA; MACIEL; MATOS, 2015)) usaram alguma técnica de medição para realizar o experimento. Por outro lado, poucos são os trabalhos que abordam a simulação em ambiente de computação móvel em nuvem, e cita-se neste caso o trabalho de (BAHWAIRETH et al., 2016).

**Tabela 3.1 Sumário dos trabalhos relacionados**

<b><i>Trabalho</i></b>	<b><i>Metodologia para avaliação de desempenho</i></b>	<b><i>Técnica de Medição</i></b>	<b><i>Técnica de Simulação</i></b>	<b><i>Ferramenta de coleta de métricas</i></b>	<b><i>Modelo de Simulação</i></b>
<i>Baseline</i>	X	X	X	X	X
(HAMAD; SAAD; ABED, 2010)	–	X	–	–	–
(HAMRÉN, 2012)	–	X	–	–	–
(SILVA; MACIEL; MATOS, 2015)	–	X	–	X	–
(BAHWAIRETH et al., 2016)	–	–	X	–	–

Neste contexto, o presente trabalho de dissertação se destaca ao propor uma metodologia para avaliação de desempenho integrada composta pelas técnicas de medição e simulação e aplicações em ambiente de computação móvel em nuvem. Indo além, o trabalho também propõe a confecção de um modelo de simulação e

apresenta uma ferramenta adaptada para a geração de carga de trabalho para ambiente de computação móvel em nuvem. Com esta proposição, espera-se que o trabalho possa contribuir na área de avaliação de desempenho em ambiente de computação móvel em nuvem, para o desenvolvimento de estudos de desempenho mais robustos considerando essas duas técnicas de avaliação de desempenho .

### **3.4. CONSIDERAÇÕES FINAIS**

Este capítulo apresentou uma visão geral do estado da arte da área de MCC por meio dos trabalhos relacionados. Para uma melhor visualização, esses trabalhos foram categorizados em: trabalhos que abordam a técnica de medição e trabalhos que abordam a técnica de simulação; por fim foi apresentada uma análise comparativa dos trabalhos. No próximo capítulo, será detalhado a metodologia de avaliação de desempenho que é composta pelas técnicas de medição e simulação. Nele, cada uma das atividades que compõe a metodologia será apresentada em detalhado. Por fim, é apresentado o modelo de simulação desenvolvido.



## 4. METODOLOGIA DE AVALIAÇÃO DE DESEMPENHO DE APLICAÇÕES EM AMBIENTES DE COMPUTAÇÃO MÓVEL EM NUVEM

É recorrente na área de computação, tanto em relação a *hardware* quanto a *software*, a realização de questionamentos como os que seguem: como realizar a comparação do desempenho de sistemas; como verificar se um determinado sistema é adequado aos requisitos estabelecidos; como fazer previsão de desempenho em função do crescimento da carga de trabalho ou demanda de uso; e como determinar uma configuração dos parâmetros de um sistema que atenda aos acordos de nível de serviço (SLA).

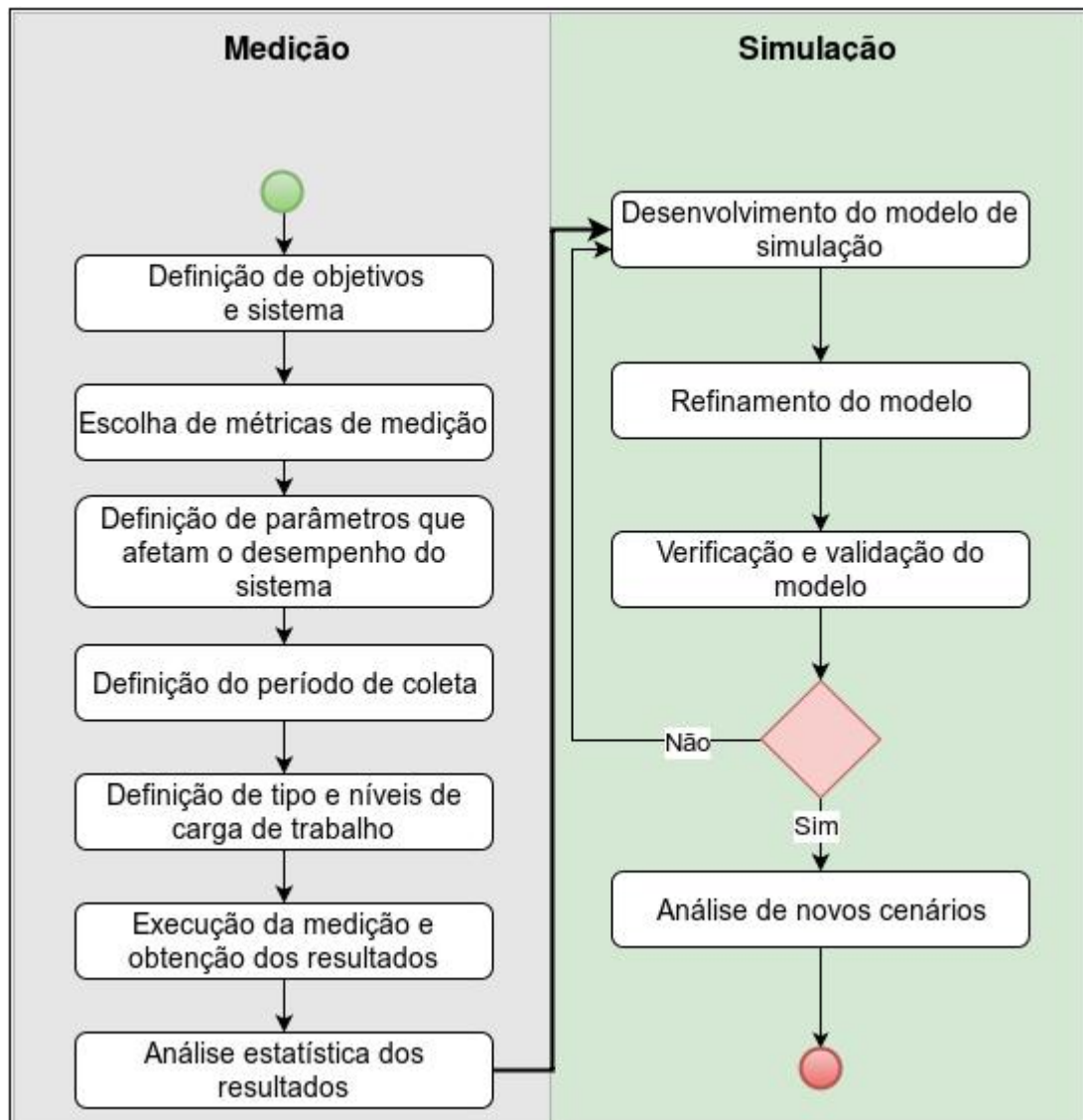
Em se tratando de *Mobile Cloud Computing* (MCC), perguntas que normalmente surgem são: seria possível estender a autonomia energética de dispositivos móveis através do uso de nuvens computacionais? Qual o percentual de ganho em termos de desempenho que a nuvem poderia oferecer? Qual o impacto da latência no desempenho de tarefas realizadas fora do dispositivo móvel?

Para dar suporte as respostas de tais questionamentos, este capítulo apresenta uma metodologia para a avaliação de desempenho de aplicativos em sistemas baseados em ambientes de computação móvel em nuvem, baseado especificamente na plataforma *Android* (GOOGLE, 2016a). Esta metodologia é composta por diversas atividades que estão agrupadas em duas fases: medição e simulação. A metodologia é apresentada na Figura 4.1. O círculo superior esquerdo representa o início do fluxo. Os retângulos com bordas arredondadas representam as atividades. O losango representa uma decisão. Por fim, o círculo direito inferior representa o fim da metodologia.

Na Figura 4.1, pode-se observar que a metodologia é dividida em duas fases. A primeira fase se dá pela avaliação de desempenho por meio da técnica de medição. A abordagem pela técnica de medição consiste em realizar coleta de métricas de forma direta no sistema real e em seguida fazer análises estatísticas com base nos dados coletados. Já a segunda fase é realizada via simulação permitindo, por exemplo, o estudo de novos cenários, com parâmetros diferentes

dos aplicativos originalmente no experimento de medição. Nesse ponto, o modelo de simulação será desenvolvido a partir dos dados e distribuições que foram geradas por meio da fase de medição, tendo-se como resultado um modelo de simulação com acurácia adequada.

**Figura 4.1 Metodologia de avaliação de desempenho de aplicativos em ambiente de computação móvel em nuvem**



Fonte: Adaptado de (JAIN, 1991)

Nas seções seguintes, cada atividade da metodologia de avaliação de desempenho será apresentada. Inicialmente serão apresentadas as atividades

relacionadas à fase de medição e em seguida serão descritas as atividades relacionadas à fase de simulação.

#### **4.1. FASE DE MEDIÇÃO**

Nessa seção será apresentada a primeira fase da metodologia que consiste em avaliar o desempenho do sistema por meio da abordagem de medição. A técnica de avaliação de desempenho por medição pode ser dividida em três partes. A primeira parte consiste no planejamento do experimento e é composto das seguintes atividades: Definição de sistema e objetivos; Escolha de métricas; Definição de parâmetros que afetam o desempenho do sistema; Definição do período de coleta; Definição de tipo e níveis de carga de trabalho. A segunda consiste da execução do experimento e é composta pela atividade de execução da medição e obtenção dos resultados. A terceira parte é composta apenas da atividade de análise estatística dos resultados obtidos por meio do experimento.

##### **4.1.1. DEFINIÇÃO DE OBJETIVOS E SISTEMA**

A primeira atividade da metodologia proposta consiste na definição dos objetivos do estudo e do sistema. Para realização desta atividade é necessário o pleno entendimento do funcionamento e da arquitetura do ambiente. Perguntas como as que seguem exemplificam o que é importante neste entendimento: quais são os requisitos do sistema? Qual é o tipo de *software* ( sistema operacional, banco de dados), quais serviços/aplicações serão usados no sistema? Tais questionamentos servem para descrever de forma sistemática a arquitetura do sistema e seus componentes. Esse estudo prévio proporciona rever questões relacionadas ao desempenho, como por exemplo, a escolha de um banco de dados por outro que esteja mais alinhado com o objetivo do sistema.

Após o estudo inicial do ambiente, a escolha dos objetivos e do sistema irá nortear todo o projeto. Por ordem lógica, primeiro se define os objetivos do estudo, ou o que precisa ser avaliado no ambiente. O segundo passo consiste na definição do sistema ou escopo do sistema em que o estudo será realizado.

Por exemplo, ao se escolher como objetivo a comparação da capacidade de processamento de dois *smartphones* distintos, o sistema a ser estudado consiste do processador, rotinas internas, barramento do dispositivo móvel, dentre outras características. Métricas diretamente associadas ao processador, neste caso, devem ser priorizadas.

Outro exemplo de objetivo no contexto de MCC é a determinação do desempenho de uma aplicação tanto em termos de tempo de execução de tarefas, quanto em consumo de energia. Por exemplo, verificar se a execução em nuvem é mais rápida que a execução local (no próprio dispositivo móvel) ou mais eficiente em termos energéticos.

#### **4.1.2. SELEÇÃO DE MÉTRICAS**

Após a definição dos objetivos do estudo e do sistema é necessário entender como a avaliação de desempenho deve ser efetivamente mensurada. Em se tratando de sistemas computacionais, as características básicas são normalmente expressas por meio de medidas como: contagem (número de ocorrências de um determinado evento), duração (intervalo de tempo do evento) e tamanho dos parâmetros. Dependendo do objetivo do estudo, tais medidas podem vir a ser usadas como métricas. Por exemplo: qual o número processos realizados em uma hora por uma *CPU*? Nesse exemplo o objetivo é apenas a medida do número de processos realizados. Por outro lado se a pergunta fosse: qual o número médio de processos realizados em uma hora por uma *CPU*? Nesse cenário, o resultado passa a ser a razão entre o número de processos executados e o tempo total (em horas) gasto para executá-los. Como regra geral, tem-se que uma métrica de desempenho é a razão de uma medida de interesse em relação a outra, normalmente a medida de tempo (LILJA, 2000).

A seleção da métrica consiste de um trabalho minucioso além de ser uma atividade chave na avaliação de desempenho. Algumas das características básicas que as métricas devem ter para que representem adequadamente o funcionamento do sistema são: **linearidade** – se o valor de uma métrica aumentar linearmente, o desempenho do sistema também deve aumentar linearmente; **confiabilidade** - se

uma métrica x diz que o sistema A é melhor que o sistema B, então o resultado do sistema A deve ser realmente melhor que o de B, caso contrário a métrica não é confiável; **consistência** – o resultado produzido por um sistema deve ser igual ao produzido por outros; e **facilidade de aferição** – se uma métrica é difícil de coletar, é improvável que se torne uma métrica muito usada (LILJA, 2000).

Em seguida, para selecionar de fato as métricas existem algumas técnicas como por exemplo a listagem de serviços do sistema. Essa atividade consiste em elencar os serviços presentes no sistema e as saídas que são geradas. A partir desta observação, é possível identificar se a saída gerada por um dado serviço satisfaz as características básicas de uma métrica de desempenho.

Outra técnica para a seleção de métricas é uso de ferramentas de coleta de métricas, bastando assim verificar quais métricas podem ser coletadas. Por exemplo, em um ambiente de servidores *Linux* ou *Windows*, existem *softwares* que geram métricas para quase todos os recursos do sistema, como: rede, CPU, memória RAM. Entretanto, em um ambiente de computação móvel, e em específico a plataforma *Android*, poucas são as alternativas de ferramentas para o estudo de desempenho do sistema. No caso da métrica de interesse não ser monitorada por nenhuma das ferramentas encontradas, uma solução seria o desenvolvimento de um programa para a coleta da métrica de interesse.

Apesar da caracterização das métricas de desempenho, quais seriam as principais métricas em um ambiente de computação móvel em nuvem? Em termos gerais, pode-se afirmar que os principais recursos computacionais presente nos dispositivos móveis são: rede, CPU, GPU, memória RAM (PANDI; CHARAF, 2013). É importante ressaltar também que cada recurso possui seu próprio conjunto de métricas e que podem ser incluídas na avaliação de desempenho.

Como exemplo, as principais métricas associada à rede são: taxa de transferência de dados (dados entregues por unidade de tempo); latência (tempo gasto no envio de pacote); taxa de perda de pacotes e a taxa de erros (número pacotes com erro em relação ao total de transmissão). Já as métricas associadas a CPU são: o tempo de utilização da CPU (tempo de execução da aplicação na CPU);

o número de operações de leitura/escrita na memória RAM; o acesso a periféricos externos e a taxa de tarefas executadas por unidade de tempo.

Em adicional as métricas apresentadas, existem um conjunto de métricas comumente usadas na avaliação de desempenho. Tais métricas não estão relacionadas ao desempenho de apenas um recurso, mas do sistema como um todo. As métricas são o tempo de resposta e o *throughput* (JAIN, 1991). Pode-se definir o tempo de resposta como sendo o intervalo de tempo entre o início de uma requisição e a resposta dada pelo sistema. Já o *throughput* pode ser definido como uma taxa do número de requisições que serão enviadas para o sistema por unidade de tempo.

#### **4.1.3. DEFINIÇÃO DE PARÂMETROS QUE AFETAM O DESEMPENHO DO SISTEMA**

A atividade definição de parâmetros consiste no levantamento dos fatores que afetam ou que podem afetar diretamente o desempenho do sistema. Para uma melhor organização, a listagem pode ser dividida em duas categorias de parâmetros: carga de trabalho e do sistema em estudo (JAIN, 1991). Os parâmetros do sistema são compostos de *software* e *hardware* e dependendo do contexto, podem interferir no experimento. Essa interferência ocorre em contextos no qual *hardware* ou *softwares* distintos são usados. Por exemplo, o *hardware* pode interferir no experimento em contextos que exista diferença no desempenho de recursos computacionais como CPU e memória RAM. No contexto de computação móvel em nuvem, os principais parâmetros do sistema são rede (3G, 4G, Wi-Fi), poder computacional da CPU/GPU e Memória RAM. Já os parâmetros da carga de trabalho, tendem a ser ajustados para caracterizar contextos em que o usuário utilize o sistema.

#### **4.1.4. DEFINIÇÃO DE PERÍODO DE COLETA**

Esta atividade deve responder questionamentos como, por exemplo: qual deve ser o período de coleta (exemplo: 1h)? O tempo de coleta será em função da carga de trabalho aplicada no experimento? A coleta será realizada em um único dia ou vários dias, por em um período específico ou aleatório? Ou seja, o período de

coleta deverá ser definido baseado no sistema e nos objetivos elencados para o projeto de avaliação de desempenho. Outro fator que deve ser levado em consideração ao definir o período de coleta são as características da métrica selecionada. Algumas métricas precisam de mais tempo que outras devido a maior variabilidade dos dados coletados.

Como exemplo, o período de coleta um servidor de um sistema web é normalmente entre o horário das 7h às 23h, uma vez que esse é o período de maior tráfego de dados e de maior número de usuários.

#### **4.1.5. DEFINIÇÃO DE TIPO E NÍVEIS DE CARGA DE TRABALHO**

A carga de trabalho consiste em gerar tarefas para serem executadas no sistema no intuito de emular situações reais que podem ocorrer. Ou seja, ela deve fazer com que o sistema funcione similarmente ao ambiente real (JAIN, 1991). A carga de trabalho deve ser representativa para o sistema real.

Nesse contexto, pode-se dividir a atividade de definição de carga de trabalho em duas etapas. A primeira etapa consiste da definição do tipo de carga que será aplicada. Ou seja, dado um sistema computacional, qual seria tipo de carga mais adequado para ser aplicada? A resposta para a pergunta se dará pelos objetivos, pelas métricas selecionadas. Se o objetivo da avaliação de desempenho for a velocidade de processamento de um sistema operacional e/ou processador para cálculos matemáticos, a carga de trabalho a ser aplicada poderia ser *benchmarks* para cálculo de números inteiros e de ponto flutuante como: o *Whetstone*, *Drystone* (WEICKER, 1990). Ou seja, o tipo de carga de trabalho está diretamente relacionado ao sistema. No caso, se o sistema agora fosse um banco de dados, as operações do *benchmark* seriam operações de escrita e leitura no banco de dados.

Em se tratando de computação móvel em nuvem, a carga de trabalho deverá seguir as mesmas regras. A partir do objetivo escolhido é que a carga de trabalho será definida. Desta forma, se o objetivo fosse avaliar o desempenho de operações em imagens como inverter e recortar no ambiente de computação móvel em nuvem, a carga de trabalho deverá ser uma ferramenta/*benchmark* que execute tais operações com imagem.

A segunda etapa, consiste da definição do nível de intensidade da carga de trabalho. Para essa atividade, uma ferramenta que pode ser usada vem a ser o *Design of Experiments* (DOE) ou planejamento de experimentos, (JAIN, 1991)(LILJA, 2000). O DOE é uma ferramenta que pode ser usado em várias situações na qual seja necessário planejar um experimento. Ele permite que múltiplos fatores de entrada sejam estudados de acordo com a saída gerada. Em virtude do DOE ser uma ferramenta desenvolvida para a realização de experimento com múltiplos fatores, é possível explorar combinações dos fatores sistematicamente de forma a ter uma visão geral do comportamento do sistema. Ou seja, é possível testar praticamente as possibilidades de um sistema para um conjunto de fatores. Isto é possível pois com o DOE, pode-se trabalhar com todas as combinações de entrada (*full factorial*) ou apenas parte das combinações (*fractional factorial*). O DOE pode ser usado quando um experimento possui mais de uma entrada e se suspeita que elas influenciam no resultado ou se existe alguma relação entre as entradas (LILJA, 2000) (JAIN, 1991)(SEMATECH, 2006).

No intuito de ilustrar o planejamento de um experimento usando a metodologia do DOE, segue um exemplo. O objetivo do estudo visa avaliar o desempenho de um dispositivo móvel (com sistema operacional *Android*) em relação a operações com imagem. A carga de trabalho consiste de um programa que realiza operações de redimensionamento de imagens e recebe dois fatores como entrada. Os fatores são a qualidade e o tamanho da imagem redimensionada final. Como existem ao menos dois fatores, a metodologia DOE pode ser aplicada.

Sendo assim, a primeira etapa do DOE consiste em listar os fatores existentes no sistema e neste caso são a qualidade e o tamanho da imagem que será redimensionada. Como são apenas dois fatores a abordagem de combinação de fatores do DOE usada será o *full factorial*. Entretanto, em experimentos com número alto de fatores, pode-se optar pela abordagem *fractional factorial* e diminuir o número de combinações. A segunda etapa consiste em definir os níveis dos fatores que segundo o DOE só precisam ser dois, o nível mínimo e o nível máximo (SEMATECH, 2006). A definição do nível mínimo e máximo deve ser realística de acordo com o sistema. Não se recomenda definir níveis com valores que seriam



inatingíveis no sistema real. Para fins didáticos, se assumirá que o nível mínimo será representado por -1 e o nível máximo será representado por +1. As combinações possíveis para os 2 fatores com dois níveis são 4 e estão apresentadas na Tabela 4.1.

A Tabela 4.1 apresenta as quatro combinações de experimentos possíveis, e esta técnica permite explorar todas as possibilidades no sistema para o nível mínimo e máximo de cada fator. Caso necessário, é possível trabalhar com um número maior de níveis, como um nível médio entre o mínimo e o máximo (SEMATECH, 2006).

**Tabela 4.1 Combinações possíveis para dois fatores usando *Full Factorial***

<b>Experimento</b>	<b>Qualidade image</b>	<b>Tamanho Imagem</b>
1	-1	-1
2	-1	+1
3	+1	-1
4	+1	+1

#### **4.1.6. EXECUÇÃO DA MEDIÇÃO E OBTENÇÃO DOS RESULTADOS**

A presente atividade consiste em tornar factível as decisões tomadas até então, como definição de sistema, escolha de métricas, escolha de carga de trabalho. Será nesta atividade que as ferramentas de coleta serão configuradas/instaladas de forma que o experimento de avaliação possa ocorrer. Como exemplo, para se coletar métricas do sistema operacional *Linux*, é necessário configurar e/ou instalar os programas (como exemplo: *iostat* – monitora de métricas de entrada e saída (disco); *IPTraf* – monitora estatísticas de rede) que serão usados para coletar métricas e a carga de trabalho para simular o estresse do sistema.

Outra parte importante desta atividade são algumas precauções que devem ser tomadas como: realizar o isolamento do sistema no intuito de diminuir as chances de ruídos nas métricas coletadas; assegurar que os experimentos serão executados nas mesmas condições, mesmo que os sistemas sejam diferentes (como é o caso da temperatura, quando o sistema vem a ser *hardware*).

Como exemplo de isolamento no contexto de computação móvel, ao se estudar a taxa de transmissão de dados de um dispositivo móvel para um servidor é necessário que no momento da realização do experimento, apenas a aplicação em teste esteja transmitindo dados para o servidor, do contrário a coleta de métricas poderá não ser precisa.

#### 4.1.6.1. FERRAMENTAS PARA COLETA DE MÉTRICAS EM DISPOSITIVOS MÓVEIS

A seguir, são apresentadas algumas das ferramentas disponíveis na plataforma *Android* e que permitem a coleta de métricas.

**Systrace** – *Software* de código aberto que é utilizado para a análise de desempenho de aplicações na plataforma *Android* (GOOGLE, 2016a). Alguns dos recursos que podem ser monitorados com o *Systrace* são os processos, memória, CPU, operações de escrita e leitura, operações gráficas, exceto rede. Embora o *Systrace* realize a coleta de métricas de desempenho, ele foi desenvolvido para fins de depuração da aplicação. A depuração nesse caso ocorre por meio de um cabo USB, no qual o celular é ligado a um computador. Desta forma, o uso do *Systrace* implica na execução da aplicação mais lenta que o normal, uma vez que existe um agente externo (computador) é conectado fisicamente ao dispositivo móvel. Em um contexto de avaliação de desempenho, o *Systrace* pode inclusive interferir nos resultados devido ao fato de seu funcionamento intrusivo poder gerar perturbação na avaliação no sistema *Android*.

**Emmagee** – É uma ferramenta para coleta de métricas de desempenho do sistema operacional *Android*. As métricas disponíveis são memória, tráfego de rede, consumo da CPU (aplicação e do sistema), estado e consumo da bateria. Esta ferramenta gera um arquivo de *log* das métricas coletadas no formato csv. Esse arquivo provê suporte para a realização da análise posterior dos dados coletados (LINARES-VÁSQUEZ et al., 2015) (NETEASE, 2016).

**Cloudbench** – É um aplicativo para *Android* que implementa um conjunto de *benchmarks*. O objetivo do *CloudBench* é investigar por meio de um estudo quantitativo, se um celular é mais rápido usando a “nuvem” em termos de tempo de

resposta em comparação com a tarefa sendo executada no dispositivo móvel. Para a realização do estudo, foram desenvolvidas três tarefas para serem executadas localmente e em seguida na nuvem. As tarefas são: ordenação de listas, computar a quantidade de números primos até um valor limite e operações de redimensionamento e rotação com imagens (HAMRÉN, 2012). O *Cloudbench* possui as limitações de gerar arquivos de logs das métricas coletadas bem como não implementa um mecanismo que permita definir o tempo entre envios para estressar o sistema.

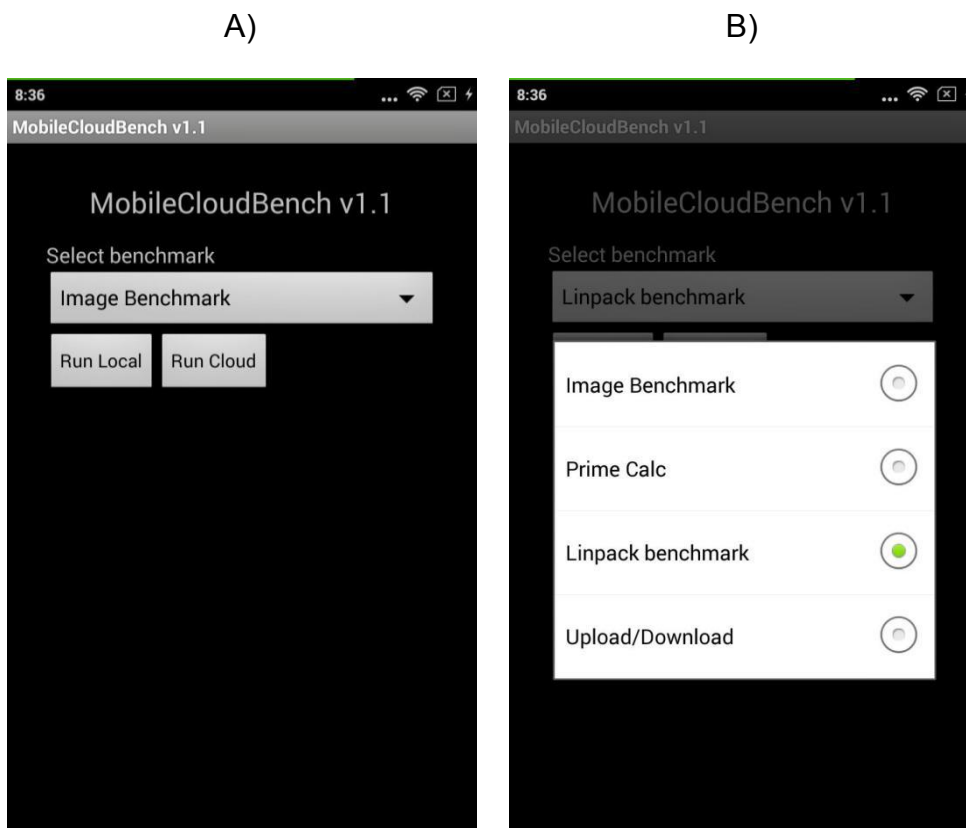
#### **4.1.6.1. FERRAMENTA PROPOSTA**

Dentre os aplicativos apresentados, o aplicativo que estava mais alinhado com a proposta e os requisitos do experimento foi o *CloudBench*. Contudo não era possível usá-lo nos experimentos, pois, ele foi desenvolvido para funcionar apenas para o *PaaS* do Google App Engine (GOOGLE, 2016b). A partir deste cenário, as opções eram: implementação da *software* do zero para os objetivos de geração de carga no dispositivo móvel ou usar o aplicativo *Cloudbench* como base de partida para o desenvolvimento da proposta. Optou-se por usar o *Cloudbench* como ponto de partida, e as modificações foram aplicadas.

Inicialmente a primeira modificação no *CloudBench* foi a de permitir o uso de outras plataformas de nuvem além do App Engine (GOOGLE, 2016b), como a da Amazon (AMAZON, 2016) e da Microsoft Azure (MICROSOFT, 2016). Neste caso também foi necessário implementar a aplicação do servidor que receberá as tarefas do dispositivo móvel e que representará a nuvem. Outra modificação importante foi implementar na aplicação um agendador de execução de tarefas por meio de um tempo fixo como, por exemplo, 0,05 segundos (neste caso, as tarefas serão enviadas a cada 0,05 segundos). Em adicional foi implementado o *benchmark Linpack* no *Cloudbench*, o qual foi usado no estudo de caso do presente trabalho. Por fim, foi implementado também um gerador de arquivos de *log* com a coleta de métricas como tempo de resposta, tempo de execução, tempo de espera na fila.

A Figura 4.2 apresenta a interface gráfica da aplicação *MobileCloudBench*, desenvolvida na plataforma Android para a geração de carga de trabalho e coleta de métricas necessárias para a realização do estudo de avaliação de desempenho.

**Figura 4.2 Ferramenta de avaliação de desempenho proposta**



A partir da interface gráfica, é possível escolher qual será o benchmark usado no teste e em que ambiente ele será executado, na nuvem ou no próprio dispositivo móvel.

#### **4.1.6.2. COMPARATIVO DAS FERRAMENTAS APRESENTADAS**

A partir das ferramentas listadas, na Tabela 4.2 é apresentado um comparativo em termos de funcionalidades das ferramentas citadas.

**Tabela 4.2 Comparação de ferramentas de coleta de métricas no sistema *Android***

	<b>Systrace</b>	<b>Emmagee</b>	<b>CloudBench</b>	<b>Adaptação do CloudBench (MobileCloudBenc)</b>
Tipo de ferramenta	Monitorar	Monitorar	Benchmark	Benchmark/ Monitorar
Monitora Memória RAM	Sim	Sim	Não	Não
Monitora CPU	Sim	Sim	Não	Não
Monitora Bateria	Sim	Sim	Não	Não
Monitora Tráfego de Rede	Sim	Sim	Não	Sim
Monitoramento de componentes gráficos	Sim	Não	Não	Não
Tempo de resposta	Não	Não	Sim	Sim
<i>Tempo em fila</i>	<i>Não</i>	<i>Não</i>	<i>Não</i>	<i>Sim</i>

Na Tabela 4.2 a ferramenta adaptada (MobileCloudBenc) apresenta algumas vantagens em relação ao CloudBench, como por exemplo a coleta do "tempo em fila". Contudo, não é possível comparar MobileCloudBenc com o Systrace ou o Emmagee pois são tipos distintos de aplicações.

#### **4.1.7. ANÁLISE ESTATÍSTICA DOS RESULTADOS**

Em relação a análise, é importante ressaltar a existência de variação nas métricas coletadas em cada experimento realizado, assim como espera-se resultados que sejam considerados discrepantes ou *outliers*. *Outliers* são valores que são significativamente diferentes em relação aos outros resultados observados.

No primeiro momento é necessário aplicar um filtro de forma a remover os resultados discrepantes, evitando assim conclusões baseadas em dados inconsistentes. No segundo momento, para a realização das comparações e análises de desempenho é essencial levar em consideração a variação dos dados coletados e é bastante comum a utilização de técnicas estatísticas para a realização de comparações. Algumas ferramentas estatísticas que podem ser usadas são intervalo de confiança e testes de hipótese.

A partir da análise dos dados coletados, inicia-se a fase que consiste em compilar os resultados para serem apresentados. O ideal é que os resultados sejam apresentados com conteúdo de fácil entendimento e sem ambiguidades. E para atingir esse objetivo normalmente são usados gráficos estatísticos.

### **4.2. FASE DE SIMULAÇÃO**

Esta seção apresentará a fase de simulação da metodologia descrita na

Figura 4.1, e é composta pelas atividades: Desenvolvimento do modelo de simulação; Refinamento do modelo; Validação e verificação do modelo; Análise de novos cenários.

#### **4.2.1. DESENVOLVIMENTO DO MODELO DE SIMULAÇÃO**

O desenvolvimento do modelo de desempenho possui duas etapas importantes. A primeira vem a ser a prototipação do modelo na ferramenta ou formalismo desejado. Por exemplo, na ferramenta de simulação Arena (“Arena Simulation Software,” 2016), a prototipação do modelo é feita usando um formalismo gráfico. O segundo vem a ser a parametrização do modelo de forma a fazer com que o modelo represente um determinado sistema.

O desenvolvimento do modelo de simulação se baseia nas análises estatísticas e distribuições das métricas coletadas na primeira fase da metodologia. É importante ressaltar que, no desenvolvimento de um modelo de desempenho representativo, é essencial o entendimento do ambiente em vários aspectos (como sistema operacional, *hardware*, aplicações usadas, etc).

Para o desenvolvimento do modelo, poderão ser usados tanto modelos analíticos (Redes de *Petri* (MOLLOY, 1982), Cadeias de *Markov* (BOLCH et al., 2006)) quanto ferramentas de modelagem e simulação discreta como o Arena (KELTON; SADOWSKI; SADOWSKI, 2002).

#### **4.2.2. REFINAMENTO DO MODELO**

Após o desenvolvimento do modelo de desempenho inicial, ao se usar a técnica de modelagem escolhida, entra-se na fase de refinamento do modelo no intuito de torná-lo mais acurado em relação ao sistema real, isto é, que o resultado das simulações represente o sistema real.

Nessa atividade de refinamento, as modificações são realizadas baseadas na fase anterior ("Análise estatísticas dos resultados"), ou seja, a partir das métricas coletadas na fase de medição, são gerados os parâmetros necessários para o refinamento. Vale ressaltar que os dados coletados já passaram por filtros de *outlier* na fase anterior.

Como exemplo de parâmetro, têm-se distribuições de probabilidades traçadas baseados nas métricas monitoradas. A distribuição de probabilidade é uma ferramenta estatística que permite descrever o comportamento que uma variável poderá assumir ao longo de um espaço de valores. Para traçar as distribuições de probabilidade existem várias ferramentas, na qual se destaca o *Minitab* (MINITAB, 2016) e o *Input Analyser* do Arena ("Arena Simulation Software," 2016).

#### **4.2.3. VERIFICAÇÃO E VALIDAÇÃO DO MODELO**

Esta atividade tem como foco a verificação e validação tanto da especificação do modelo em si, quanto dos resultados produzidos por ele. A verificação é o processo de se determinar se um modelo desenvolvido faz exatamente o que foi

projetado para fazer. A verificação não trata se o modelo está correto, mas sim se ele foi implementado de forma correta (LILJA, 2000). Dentre as várias técnicas usadas para verificar um modelo, pode-se destacar como método de verificação: a verificação passo a passo ou *debug* do sistema; usar casos de usos simplificados para verificar sem precisar realizar a simulação; teste de continuidade – teste de continuidade consiste em executar o experimento várias vezes com pequenas modificações na entrada e espera-se que os resultados apresente variação linear e constante, do contrário o modelo não está validado.

Segundo (MENASCE et al., 2004), algumas perguntas que devem ser respondidas nessa atividade são: é o modelo certo para o sistema em estudo? O modelo captura o comportamento de componentes críticos do sistema real?

A validação do modelo é o processo de assegurar que o modelo produza resultados que represente o sistema real. Várias são as abordagens para a validação de um modelo, dentre as quais pode-se destacar: comparação com dados gerados pelo sistema real e comparação com modelos analíticos (SARGENT, 2005) (LILJA, 2000). Em se tratando da comparação da simulação com o sistema real, pode-se dizer que um modelo é dito validado se as métricas de desempenho (tempo de resposta, utilização de recursos) geradas na simulação tem um baixo percentual de erro em relação as medidas do sistema real. De forma geral, uma margem de erro de 10% é considerada aceitável (MENASCE; ALMEIDA; DOWDY, 1994). Já a comparação com modelos analíticos é recomendado em casos onde não é possível realizar a medição de métricas no sistema real, então usa-se um modelo analítico simplificado para validar o modelo de simulação. Na proposta da metodologia ( Figura 4.1), a validação do modelo de simulação pode ser realizada a partir das métricas coletadas na fase de medição do sistema, usando-se de ferramentas estatísticas como teste de hipótese, erro percentual relativo e intervalo de confiança.

#### **4.2.4. ANÁLISE DE NOVOS CENÁRIOS**

Dado que o modelo foi devidamente validado e apresenta resultados coerentes aos do sistema real, ele poderá ser usado para a predição do desempenho do sistema em diversos novos cenários. Como exemplos de novos



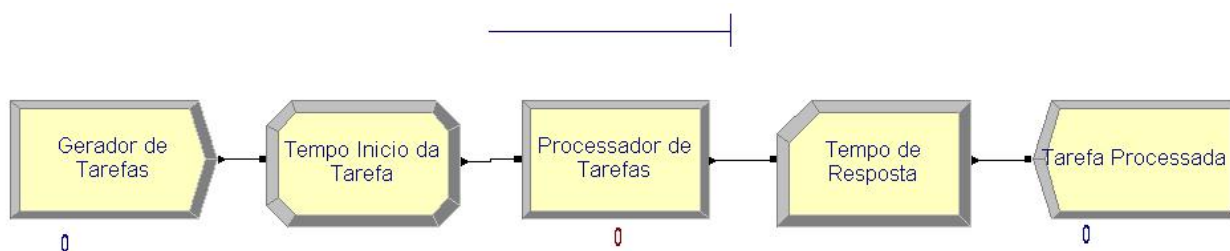
cenários pode-se citar o redimensionamento de *hardware*, mudança na intensidade da carga de trabalho do sistema, modificações na configuração do sistema. Ou seja, o modelo permite que o sistema seja avaliado em outros contextos no qual se fosse feito no sistema real poderia demandar tempo, podendo tornar o estudo inviável em termos financeiros.

Como exemplo de avaliação de novos cenários, pode-se criar novas hipóteses/cenários no qual o sistema poderia funcionar e testá-las de forma a analisar o desempenho nesse novo cenário.

### 4.3. MODELO DE SIMULAÇÃO

Esta seção apresenta o modelo de simulação para avaliação de desempenho no dispositivo móvel e na nuvem (Figura 4.3) a partir da ferramenta de simulação discreta Arena. Esse modelo simula a execução de tarefas no dispositivo móvel ou ambiente de nuvem. No modelo abaixo, o Gerador de Tarefas representa a geração de tarefas que precisam ser processadas pelo sistema; o Tempo de Início da Tarefa representa o tempo em que a tarefa foi iniciada; o Processador de Tarefas representa o tempo de processamento da tarefa e pode ser expresso por uma distribuição de probabilidade ou por um tempo constante ou ainda por agendamento; o Tempo de resposta representa o tempo final gasto pelo processador para executar cada tarefa; por fim, o módulo "Tarefa Processada" encerra as tarefas processadas.

**Figura 4.3 Modelo de simulação para avaliação de desempenho no dispositivo móvel e na nuvem computacional**



No estudo de caso, que será apresentado no próximo capítulo, o modelo será instanciado e receberá parâmetros como distribuição de probabilidade que mapeia a carga de trabalho e os tempos de resposta.

#### **4.4. CONSIDERAÇÕES FINAIS**

Este capítulo apresentou a proposta de metodologia integrada, composta por atividades de medição e simulação, para avaliação de desempenho de aplicações em ambientes de computação móvel em nuvem, e também detalhou as atividades que compõe essa metodologia. Para ilustrar e avaliar o que foi proposto neste capítulo, no próximo serão apresentados estudos de caso mostrando na prática todos os passos descritos.

## 5. ESTUDO DE CASO

O presente capítulo tem a função tanto de ilustrar na prática a utilização da metodologia como também avaliar se a mesma se mostra útil e relevante no contexto em que foi proposta. Neste sentido, o presente trabalho usará como estudo de caso o *benchmark Linpack* (DONGARRA; LUSZCZEK; PETITET, 2003). O *benchmark Linpack* consiste em uma biblioteca de sub-rotinas para a computação de operações matemáticas de álgebra linear e uma de suas principais aplicações foi a avaliação de desempenho de sistemas computacionais.

O objetivo do estudo de caso é realizar a avaliação de desempenho da execução de tarefas em dois ambientes distintos: no próprio dispositivo móvel e na nuvem computacional por meio da execução remota. Nesse contexto, tarefa é considerada a execução de programas, como *benchmark*, que possam emular situações reais no ambiente. O objetivo se mostra relevante, pois é comum a dúvida, na área de computação móvel em nuvem, se haverá um ganho no desempenho ao migrar aplicação móvel para a nuvem. Outro objetivo associado é a análise das métricas coletadas e a verificação da existência de ganhos em termos de desempenho (especificamente tempo de execução) por parte da execução remota em detrimento da execução local.

Basicamente, o sistema é constituído do dispositivo móvel, da rede pela qual os dados serão transferidos e do servidor responsável pela execução do experimento de processamento na nuvem. A visão geral do sistema é apresentada na Figura 2.1.

Figura 5.1 Visão geral do ambiente de experimentos



A nuvem (IaaS) pública no qual o experimento foi realizado foi a nuvem da Amazon em São Paulo. Usar outro tipo de plataforma como PaaS, poderá implicar em resultados distintos. As configurações do ambiente de estudo no qual foram realizados os experimentos de avaliação de desempenho são:

**Configuração do Servidor na Nuvem (IaaS):** Instância virtual do tipo *t2.medium*, localizada no *Data Center* da Amazon AWS (AMAZON, 2016) em São Paulo, com 4 GB de RAM e 2 vCPU, Intel Xeon Processors até 3.3GHz de clock, sistema operacional Ubuntu 14.04;

**Configuração do dispositivo móvel:** *smartphone* com CPU (*Snapdragon* 410 64 bit e até 1.2 GHz), memória RAM de 2GB com a versão do *android* 4.4.4 (*Kitkat*);

**Banda da rede do Servidor <-> Cliente:** 10 MB;

**Latência da rede:** 0,065 segundos

**Servidor Web/Aplicação:** Tomcat 7.0.52.0;

**Serviços Web:** Linpack;

O ambiente apresentado na visão geral acima será usado nos dois estudos de casos (*Linpack* e cálculo de números primos) que serão apresentados nas seções seguintes.

## 5.1. ESTUDO DE CASO 1: LINPACK

O estudo de caso a seguir foi executado a partir da metodologia de avaliação de desempenho proposta na Figura 4.1 e se trata de um experimento com o *benchmark Linpack* (DONGARRA; LUSZCZEK; PETITET, 2003). Inicialmente será apresentada a fase de medição e no segundo passo a fase de simulação. O *Linpack* consiste de um conjunto de sub-rotinas para a computação de operações matemáticas de álgebra linear e uma de suas principais aplicações foi a avaliação de desempenho de sistemas computacionais. A escolha do *benchmark Linpack* se deu devido ao longo histórico de uso na literatura e também por ser um *benchmark* matemático. O uso de *benchmarks* matemáticos é de grande relevância e configuram na segunda posição entre os *benchmarks* mais usados na avaliação de

desempenho de dispositivos móveis (SILVA et al., 2016). Na primeira posição estão os benchmarks de operações com imagens, contudo não foram aplicados no presente estudo de caso devido a questões técnicas na implementação do experimento.

### 5.1.1. FASE DE MEDIÇÃO

Nesta subseção será apresentada a avaliação de desempenho por meio da medição no ambiente de computação móvel e na nuvem computacional.

- **Definição de sistemas e objetivos** – O sistema a ser estudado consiste do ambiente de computação móvel em nuvem e pode ser visto na

Figura 5.1 Visão geral do ambiente de experimentos



- . O objetivo do estudo de caso consiste em comparar o desempenho na execução de tarefas no dispositivo móvel em relação a execução da tarefa na nuvem. A tarefa nesse contexto consiste na execução do *benchmark Linpack*.
- **Seleção de métricas** – As métricas de desempenho usadas no experimento foram: tempo de execução e tempo de resposta. O tempo de execução consiste no intervalo de execução do *benchmark Linpack*, sem considerar tempo de fila. Já o tempo de resposta é o intervalo de tempo entre o início de uma requisição e a resposta dada pelo sistema e neste caso o tempo de fila é considerado.

A título de explicação, a Tabela 5.1 apresenta exemplos de métricas para o tempo de execução e tempo de resposta. Percebe-se que o tamanho da fila impacta apenas no tempo de resposta. Já a coluna com tempo de

execução apresenta observações similares, tendo em vista que considera apenas a execução da tarefa, sem considerar a fila.

Tabela 5.1 Exemplo de coleta de métricas de tempo de resposta e tempo de execução

<i>Experimento</i>	<i>Tempo execução</i>	<i>Tempo resposta</i>	<i>Tamanho da fila</i>
1	6,168	6,168	18
2	6,14	12,252	17
3	6,142	18,332	16
4	6,1	24,383	15
5	6,147	30,48	14
6	6,139	36,568	13
7	6,196	42,715	12

A métrica de "Tempo execução" representa o tempo que cada atividade demora a ser realizada. A métrica de "Tempo de resposta" representa o tempo de cada atividade e considera também o tempo da fila.

- **Definição de parâmetros que afetam o desempenho do sistema** – Duas são as classes de parâmetros que podem afetar o desempenho do sistema: os parâmetros de sistema (*hardware, software*) e os de carga de trabalho.

Os parâmetros do sistema usados no experimento foram:

**Configuração do servidor:** *t2.medium*, localizada no *Data Center* da Amazon AWS em São Paulo, com 4 GB de RAM e 2 vCPU, Intel Xeon Processors até 3.3GHz de clock, sistema operacional Ubuntu 14.04;

**Configuração do dispositivo móvel** - *smartphone* com CPU (*Snapdragon 410* 64 bit e até 1.2 GHz), memória RAM de 2GB com a versão do *android 4.4.4 (Kitkat)*;

**Rede:** Conexão *WiFi* de 10MB;

O parâmetro da carga de trabalho é o parâmetro do *benchmark Linpack* que será detalhado na atividade "Definição de tipo e níveis de carga de trabalho".

- **Definição de período de coleta** – A definição do período de coleta no experimento foi definida em função da carga de trabalho gerada no sistema. Isto é, se a carga de trabalho planejada para o experimento demorar 1 hora, esse será o período que os dados serão coletados. Por outro lado, se a execução do experimento dura-se 24 horas, a coleta das métricas seria realizada no período de 24 horas.
- **Definição de tipo e níveis de carga de trabalho** – A carga de trabalho que será usada no experimento será o *benchmark Linpack*. O *Linpack* recebe como parâmetro um valor inteiro que representa o tamanho de uma matriz quadrada na qual serão realizadas operações matemáticas de ponto flutuante. Os níveis de carga de trabalho definidos para serem aplicados ao *benchmark linpack* foram 1000 e 2000. A escolha de tais níveis de carga foi baseada na metodologia do *Design of Experiments* (DOE), o qual recomenda o uso de 2 níveis de carga, um mínimo e um máximo e que sejam valores realistas em relação ao sistema (SEMATECH, 2006).

O número de requisições foi definido em 20, após um estudo de determinação do tamanho da amostra via teste de hipótese com 95% de confiança (JAIN, 1991)(SEMATECH, 2006). Nesse estudo, verificou-se que o desvio padrão para uma amostra com 500 observações era baixo e sugeria amostras com número de observações abaixo de 4. O estudo foi feito tanto para o experimento sendo executado localmente quanto na nuvem computacional. A Tabela 5.2 apresenta a configuração da carga de trabalho do experimento.

Tabela 5.2 Carga de trabalho

<i>Experimento</i>	<i>Número de requisições/ execuções</i>	<i>Tempo entre Envios de Tarefa (segundos)</i>	<i>Ambiente</i>	<i>Cargas de trabalho ( mínimo e máximo)</i>
1	20	0,05	Local e nuvem	1000; 2000;
2	20	0,1	Local e nuvem	1000; 2000;
3	20	0,5	Local e nuvem	1000; 2000;
4	20	0,75	Local e nuvem	1000; 2000;
5	20	1	Local e nuvem	1000; 2000;
6	20	1,25	Local e nuvem	1000; 2000;
7	20	1,5	Local e nuvem	1000; 2000;
8	20	2	Local e nuvem	1000; 2000;
9	20	2,2	Local e nuvem	1000; 2000;
10	20	2,5	Local e nuvem	1000; 2000;

- **Execução da medição e obtenção dos resultados** – Como apresentado na metodologia, as principais funções dessa atividade são assegurar que o experimento seja executado em um ambiente representativo em relação ao sistema real, além de tomar algumas precauções, como o isolamento do sistema. No experimento, cada experimento foi executado com intervalo de 30 segundos, no intuito de diminuir a variabilidade dos dados coletados. Esse intervalo de tempo permite que os recursos alocados para o experimento anterior sejam desalocados para a execução do experimento subsequente.

A medição das métricas do experimento ocorreu por meio do uso da ferramenta MobileCloudBench, que também foi usada para a geração da carga de trabalho no dispositivo móvel.

- **Análise estatística dos resultados** – Nesta atividade os dados coletados foram analisados usando medidas de dispersão como variância e desvio padrão, no intuito de verificar se os dados coletados estavam coerentes. Como resultado, os dados coletados se mostraram dentro do esperado.



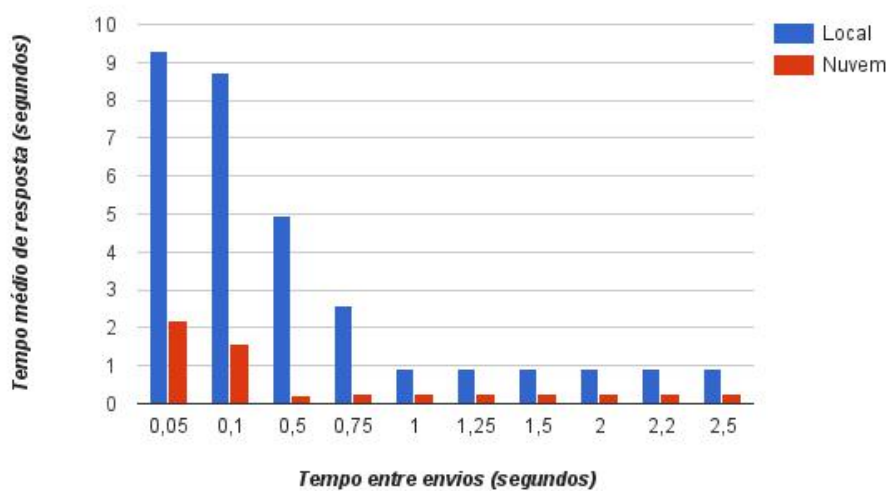
Em um segundo momento foi realizado a filtragem de observações discrepantes ou *outliers* uma vez que tais observações podem causar ruídos no resultado final (SEMATECH, 2006). Na , é apresentado o percentual de *outliers* encontrado para cada ambiente e carga de trabalho. A seguir é apresentado um comparativo do tempo médio de resposta entre a execução do *benchmark Linpack* sendo realizada no localmente (no dispositivo móvel) e na nuvem.

**Tabela 5.3 Sumarização do percentual de outliers**

<b>Carga Trabalho</b>	<b>Ambiente: Local</b>	<b>Ambiente: Nuvem</b>
1000	0,0333%	0,0250%
2000	0,0583%	0%

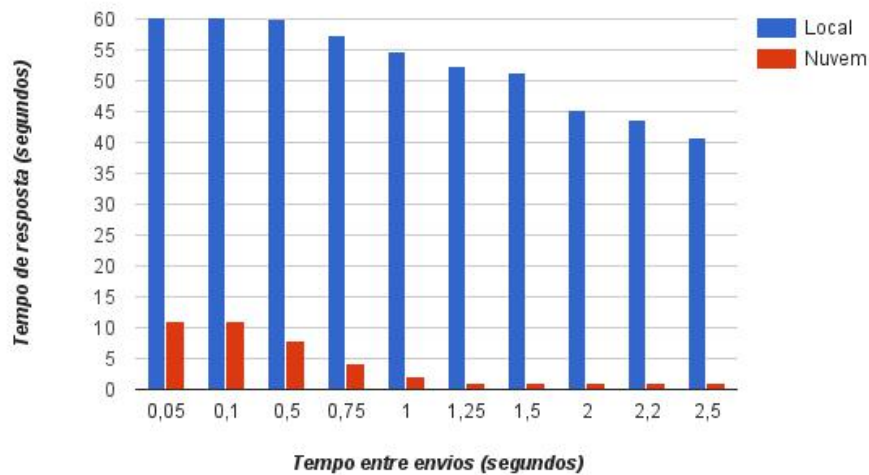
A Figura 5.2 apresenta a comparação dos resultados para o tempo médio de resposta do experimento sendo executado localmente e na nuvem. A carga de trabalho aplicada no benchmark linpack foi de 1000. Na Figura 5.2 é perceptível que o tempo médio de resposta na nuvem para as 20 requisições (definidas no estudo de caso) é menor em relação ao dispositivo móvel mesmo somado com tempo de latência da rede.

**Figura 5.2 Comparação tempo de resposta no local vs nuvem**



A Figura 5.3 apresenta a comparação dos resultados para o tempo médio de resposta do experimento sendo executado localmente e na nuvem. A carga de trabalho aplicada no *benchmark linpack* foi de 200. Nesta imagem a diferença entre o tempo médio de resposta da execução local ou na nuvem se mostra maior comparado a Figura 5.2.

**Figura 5.3 Comparação tempo de resposta no local vs nuvem.**



As análises acima mostram o quanto a nuvem pode contribuir com os dispositivos móveis para melhoria do desempenho em termos de tempos de execução.

### 5.1.2. FASE DE SIMULAÇÃO

Após o termino da fase de medição, inicia-se a fase de simulação, onde a primeira atividade consiste em desenvolver um modelo de simulação que será baseado nos resultados da fase de medição.

- **Desenvolvimento do modelo de simulação** – A ferramenta usada para a geração do modelo de simulação (Figura 4.3) foi o Arena e a primeira etapa consistiu em prototipar o modelo de acordo com o sistema. O modelo de simulação desenvolvido está descrito Seção 4.3. A próxima atividade será responsável pelo refinamento do modelo de simulação prototipado.
- **Refinamento do modelo** – O refinamento do modelo consistiu na geração distribuições de probabilidade para o tempo de execução do *benchmark*

*Linpack* para os cenários apresentados na atividade de “Definição de tipo e níveis de carga de trabalho”. Contudo, para que fosse possível realizar a validação do modelo na fase seguinte, foram selecionados apenas seis cenários para a geração da distribuição de probabilidade e os outros quatro serão usados para a validação do modelo de simulação na atividade seguinte. Os seis cenários escolhidos a partir da Tabela 5.4 foram 2, 3, 5, 7, 8 e 10 e são apresentados na Tabela 5.4.

**Tabela 5.4 Cenários usados para a geração da distribuição de probabilidade**

<b>Cenário</b>	<b>Requisições</b>	<b>Tempo entre Envio de Tarefas (segundos)</b>	<b>Carga de Trabalho</b>
2	20	0,1	1000; 2000
3	20	0,5	1000; 2000
5	20	1	1000; 2000
7	20	1,5	1000; 2000
8	20	2	1000; 2000
10	20	2,5	1000; 2000

Os cenários apresentados na tabela acima representam um total de 120 observações e representa a multiplicação dos 6 cenários pelas 20 requisições. A partir dos dados coletados dos experimentos baseados nos cenários apresentados na Tabela 5.4, foram geradas distribuições de probabilidades juntamente com as suas médias. Na Tabela 5.5 são apresentadas as distribuições de probabilidades necessárias para a calibração do modelo de simulação apresentado na Figura 4.3. As distribuições de probabilidade foram:

**Tabela 5.5 Distribuições de probabilidade para o *Linpack***

<b>Carga Trabalho</b>	<b>Ambiente: Local</b>	<b>Ambiente: Nuvem</b>
1000	$0,9 + ERLA(0,0037; 1,8)$	$0,21 + 0,09 X BETA(5,6; 8,59)$
2000	$6,05 + 0,18 X BETA(3,76; 4,19)$	$0,82 + 0,78 X BETA(2,24; 4,04)$

As distribuições acima serão verificadas e validadas na próxima atividade.

**Verificação e Validação do modelo** – Após o refinamento do modelo, chega-se a etapa de verificação do modelo. Como apresentado na subseção de “Verificação e Validação do modelo” da metodologia, a verificação consiste em determinar se o modelo de simulação está funcionando corretamente de acordo com que foi proposto. A técnica de verificação usada foi o teste de continuidade. Após pequenas modificações na entrada do modelo de simulação, os resultados apresentados se mantiveram estáveis e com baixa variância.

Já que o modelo foi devidamente verificado, entra-se a etapa validação do modelo de simulação. No refinamento, as distribuições de probabilidade foram desenvolvidas a partir dos cenários 2, 3, 5, 7, 8 e 10. Na validação, serão usados os cenários restantes (1, 4, 6 e 9) para validar o modelo. A ideia é usar cenários diferentes nas etapas de refinamento e validação no intuito de que o processo de validação possa ocorrer de forma independente, isto é, que os dados que já foram usados para construir o modelo não sejam usados também para verificar se o mesmo é válido. A validação consiste em comparar o resultado da simulação com os cenários que não foram usados na geração da distribuição. O esperado é que o modelo represente os cenários predefinidos e novos cenários.

Abaixo são apresentadas as tabelas com a comparação entre o resultado do tempo de resposta do sistema real e o resultado do tempo de resposta da simulação. A Tabela 5.6 apresenta os resultados da validação para o dispositivo móvel. Visualizando a coluna “% Erro relativo” na Tabela 5.6, percebe-se que o erro entre o tempo médio de resposta real e a simulação está abaixo dos 10%. Mesmo para os cenários 1, 4, 6 e 9, que não foram usados para o refinamento do modelo, o “% Erro relativo” ficou abaixo dos 10%. Com este resultado, conclui-se que modelo de simulação está validado, dentro de uma margem aceitável, para a execução local, isto é, no próprio dispositivo.

**Tabela 5.6 Validação do modelo de simulação baseado no tempo de resposta medido no sistema real em relação à simulação para o ambiente local**

		<i>Linpack Local</i>			
		Tempo de Resposta			
Cenários	Tempo	Carga de trabalho	Sistema Real	Simulação	% Erro relativo
<b>Experimento Carga de Trabalho 1000</b>					
1	0,05	1000	9,32	9,29	0,26%
2	0,1	1000	8,75	8,82	0,78%
3	0,5	1000	4,97	5,02	0,96%
4	0,75	1000	2,59	2,64	1,95%
5	1	1000	0,93	0,93	0,47%
6	1,25	1000	0,94	0,93	0,66%
7	1,5	1000	0,93	0,93	0,35%
8	2	1000	0,93	0,93	0,21%
9	2,2	1000	0,94	0,93	0,60%
10	2,5	1000	0,94	0,93	0,84%
<b>Experimento Carga de Trabalho 2000</b>					
1	0,05	2000	64.0252	63.923	0.16%
2	0,1	2000	63.5063	63.448	0.09%
3	0,5	2000	60.01025	59.648	0.60%
4	0,75	2000	57.32855	57.273	0.10%
5	1	2000	54.83515	54.898	0.11%
6	1,25	2000	52.4814	52.523	0.08%
7	1,5	2000	51.40885	50.148	2.45%
8	2	2000	45.31285	45.398	0.19%
9	2,2	2000	43.5918	43.498	0.22%
10	2,5	2000	40.69925	40.648	0.13%

Já a Tabela 5.7, apresenta os resultados da validação para os cenários de nuvem. Visualizando a coluna “% Erro relativo” na Tabela 5.7, percebe-se que o erro entre o tempo médio de resposta real e a simulação está abaixo dos 10%. Mesmo para os cenários **1, 4, 6 e 9**, que não foram usados para o refinamento do modelo, o “% Erro relativo” ficou abaixo dos 10%. Com este resultado, conclui-se que modelo de simulação está validado dentro de uma margem aceitável para o ambiente de nuvem.

**Tabela 5.7 Validação do modelo de simulação baseado no tempo de resposta medido no sistema real em relação à simulação para o ambiente de nuvem**

<b>Linpack Nuvem</b>					
				Tempo de Resposta	
Cenários	Tempo	Carga de trabalho	Sistema Real	Simulação	% Erro relativo
<b>Experimento Carga de Trabalho 1000</b>					
1	0,05	1000	2,1741	2,095	3,64%
2	0,1	1000	1,5945	1,62	1,60%
3	0,5	1000	0,23885	0,245	2,57%
4	0,75	1000	0,2491	0,245	1,65%
5	1	1000	0,249	0,245	1,61%
6	1,25	1000	0,2533	0,245	3,28%
7	1,5	1000	0,2753	0,245	10,01%
8	2	1000	0,2674	0,245	8,38%
9	2,2	1000	0,26455	0,245	7,39%
10	2,5	1000	0,266	0,245	7,89%
<b>Experimento Carga de Trabalho 2000</b>					
1	0,05	2000	11,13065	11,135	0,04%
2	0,1	2000	11,1422	10,66	4,33%
3	0,5	2000	7,78435	6,86	9,87%
4	0,75	2000	4,2873	4,485	4,61%
5	1	2000	2,0113	2,156	7,19%
6	1,25	2000	1,1036	1,126	2,03%
7	1,5	2000	1,0787	1,11	2,90%
8	2	2000	1,04315	1,11	6,41%
9	2,2	2000	1,07825	1,11	2,94%
10	2,5	2000	1,05105	1,11	5,61%

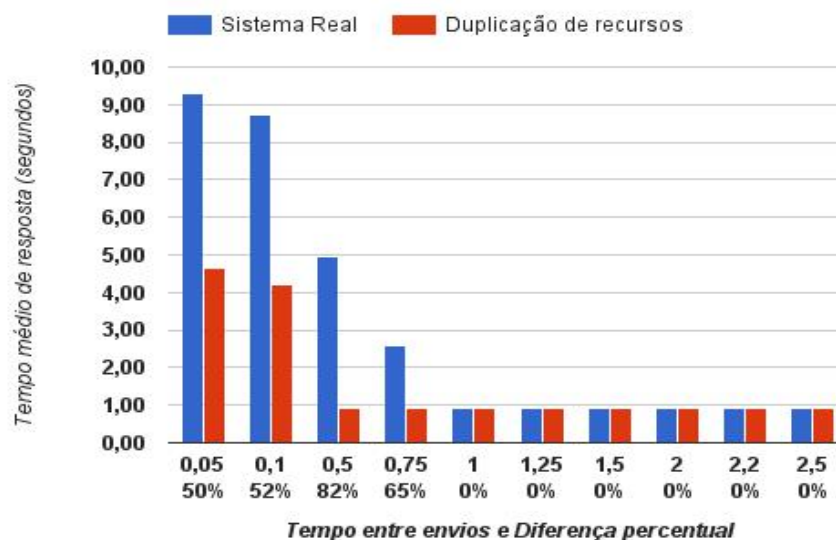
Conforme foi apresentado nas tabelas Tabela 5.6 e Tabela 5.7, o modelo de simulação se encontra devidamente validado, pois o “% Erro relativo” ficou abaixo dos 10% entre o sistema real e a simulação.

- **Análise de novos cenários** - A análise de novos cenários usando modelos de simulação permite o estudo hipóteses sem a necessidade da realização de novos experimentos de medição. A Figura 5.4 apresenta um estudo no qual os recursos de computação do dispositivo móvel e do servidor em nuvem são dobrados. O dobramento de recursos computacionais para o dispositivo

móvel consiste em uma situação hipotética. Contudo, o dobramento de recursos para um servidor/nuvem é normal e pode ser feita tanto verticalmente ( aumentando o recurso de memória e processamento) quanto horizontalmente (por meio da instanciação de dois servidores de nuvem).

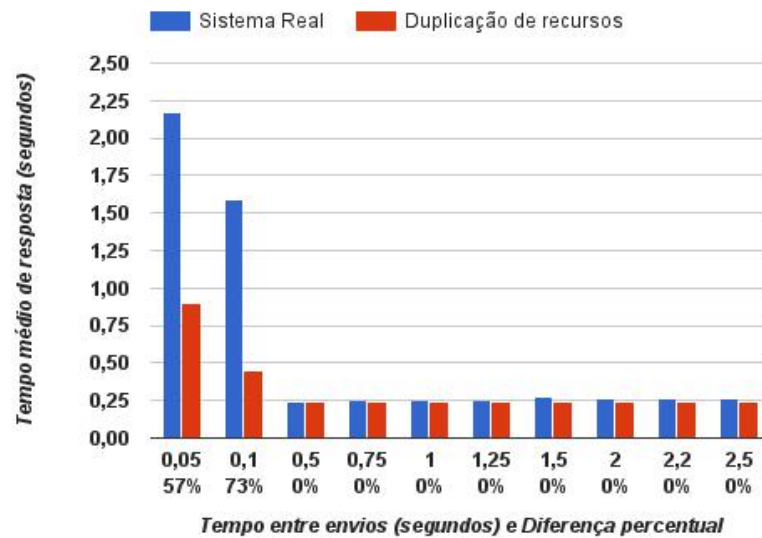
A carga de trabalho do experimento foi 1000. A simulação apresenta que no mínimo é possível ganhar em termos de desempenho, 50% para o cenário de tempo entre envios de **0,05**. Já para os cenários de tempo entre envios de **0,1**, **0,5**, **0,75**, o ganho de desempenho foi maior que 50%. Nos demais cenários, o tempo de inicialização não causa estresse no ambiente, por isso o tempo médio de resposta é constante.

**Figura 5.4 Comparação do tempo médio de resposta: sistema real vs duplicação de recursos**



A Figura 5.5 a seguir apresenta um estudo no qual os recursos de computação são dobrados para o ambiente de nuvem. A carga de trabalho do experimento foi 1000. Neste caso, dobrar os recursos de nuvem, é uma alternativa viável, uma vez que esta não tem as limitações dos dispositivos móveis. Para os cenários **0,05** e **0,1**, foi possível obter ganhos acima de **56%**. Para os demais cenários, o tempo entre envios de tarefas foi suficiente para a não geração de filas.

Figura 5.5 Comparação do tempo médio de resposta: sistema real vs duplicação de recursos



## 5.2. CONSIDERAÇÕES FINAIS

Este capítulo apresentou estudos de caso que ilustraram a aplicação da metodologia de avaliação, e assim como permitiu ilustrar a execução desta metodologia em exemplos práticos e reais e permitiu também avaliar se a metodologia foi adequada para o contexto que foi proposta. Na próxima seção, serão apresentados as conclusões, contribuições e trabalhos futuros que foram desenvolvidas a partir do presente trabalho.



## 6. CONCLUSÕES E TRABALHOS FUTUROS

### 6.1. CONCLUSÕES

A despeito da importância da avaliação de desempenho de dispositivos móveis, um fato relevante é que a realização de experimentos e coleta de métricas de desempenho no ambiente de computação móvel em nuvem possui uma série de barreiras, e uma das mais consideráveis é a falta de uma metodologia que possa dar suporte ao processo como um todo.

Tendo em vista essa problemática, o trabalho proposto apresentou uma metodologia de avaliação de desempenho inspirada em técnicas de medição e de simulação já consolidadas (JAIN, 1991; LILJA, 2000). A metodologia é composta por um conjunto de atividades destinadas a avaliação de desempenho de dispositivos móveis no contexto de computação móvel em nuvem. As atividades compreendem desde o planejamento dos experimentos da medição até o desenvolvimento do modelo de simulação e sua aplicação, bem como a análise dos resultados em novos cenários.

Em complemento, o trabalho apresentou um estudo de caso (*benchmark Linpack*) para a aplicação da metodologia. O estudo de caso objetivou avaliar se a metodologia foi adequada para a aplicação em um caso real. Através do estudo, verificou-se que o uso da metodologia foi adequado e permite que a avaliação de desempenho de computação móvel em nuvem possa ser realizada de forma sistemática.

Em se tratando das comparações relacionadas à execução das tarefas no dispositivo móvel em relação à nuvem, a nuvem se mostrou mais rápida na execução de tais tarefas. Contudo, a nuvem em certos contextos pode ter desempenho abaixo do esperado como em conexões com alta latência ou com largura de banda limitada.

## 6.2. CONTRIBUIÇÕES

A principal contribuição científica do trabalho consiste na proposta de uma metodologia para avaliação de desempenho composta pelas técnicas de medição e de simulação para uma área recente que vem a ser a computação móvel em nuvem. Em adicional, foi proposto um modelo de simulação baseado na ferramenta de simulação Arena.

A seguir são apresentadas as demais contribuições:

- Ferramenta adaptada:

Durante o projeto, algumas aplicações móveis foram avaliadas para a realização do estudo de avaliação de desempenho no sentido de geração de carga de trabalho tanto para a nuvem quanto para o dispositivo móvel. Contudo, não foi encontrado um projeto que se adequasse as especificações do experimento. Neste contexto, foi encontrada uma aplicação na plataforma *Android*, *Cloudbench* (HAMREN, 2015) no qual foi possível obter o código fonte e implementar modificações. Inicialmente a primeira modificação no *CloudBench* foi a de permitir o uso de outras plataformas de nuvem além do App Engine (GOOGLE, 2016b), como a da Amazon (AMAZON, 2016) e da Microsoft Azure (MICROSOFT, 2016). Neste caso também foi necessário implementar a aplicação do servidor que receberá as tarefas do dispositivo móvel e que representará a nuvem. Outra modificação importante foi implementar na aplicação um agendador de execução de tarefas por meio de um tempo fixo, como por exemplo, 0,05 segundos (neste caso, as tarefas serão enviadas a cada 0,05 segundos). Em adicional foi implementado o *benchmark Linpack* no *Cloudbench*, o qual foi usado no estudo de caso do presente trabalho. Por fim, foi implementado um também um gerador de arquivos de *log* com a coleta de métricas como tempo de resposta, tempo de execução, tempo de espera na fila.

Links para acessar o código fonte:

Servidor - <https://github.com/nielsonsantana/mobilecloudbenchserver>

Móvel - <https://github.com/nielsonsantana/mobilecloudbenchapp>

- Em termos de divulgação do trabalho, parte da metodologia proposta nesta dissertação foi submetida para a revista IEEE America Latina.

*N. A. SANTANA, F. A. A. LINS e E. T. G. SOUSA. Avaliação de Desempenho de Aplicações Móveis em Ambiente de Computação Móvel em Nuvem. Revista IEEE América Latina, 2016. Em revisão.*

Resumo submetido ao XV JEPEX:

*N. A. SANTANA, F. A. A. LINS. Metodologia de avaliação de desempenho de aplicações em ambiente de computação móvel em nuvem. XV Jornada de Ensino, Pesquisa e Extensão da UFRPE, 2015.*

### **6.3. LIMITAÇÕES**

Como principal limitação encontrada no trabalho está o uso de apenas uma nuvem computacional pública (AMAZON) e a de apenas um dispositivo móvel para a realização dos experimentos.

### **6.4. TRABALHOS FUTUROS**

Dentro do contexto deste trabalho, vislumbram-se diversas oportunidades de continuidade. A seguir, serão listados os principais trabalhos futuros relacionados a esta dissertação:

- Aprimorar a ferramenta proposta para a geração de carga de trabalho em ambiente de computação móvel em nuvem no sentido de criação de documentação de uso e instalação no intuito de permitir outros trabalhos possam usa-la;
- Realizar experimentos de avaliação de desempenho em novos ambientes, usando outros provedores de nuvem pública bem como usar nuvem privada local. Por fim, também testar *benchmarks* de outras categorias como *games*, *Imagem*, *stream* de vídeo (SILVA et al., 2016);
- Realizar o estudo de avaliação de desempenho considerando a otimização do consumo de bateria no dispositivo móvel, isto é, equilibrar o desempenho e o consumo da bateria dos dispositivos móveis. Este estudo é importante devido

ao fato de o dispositivo móvel ter restrições de autonomia de bateria (KUMAR; LU, 2010)(BAHL et al., 2012);

- Fazer análise de desempenho tendo em vista o paradigma de *cloudlets* que vem a ser uma “nuvem” intermediária e fisicamente próxima do usuário, em contraste com a nuvem computacional tradicional (SATYANARAYANAN et al., 2009). A proposta desta nuvem intermediária é a diminuição da latência além do aumento de banda, permitindo que as aplicações sejam mais interativas; Neste contexto, a proposta consiste em fazer estudos relacionados a distância física mínima ou número mínimos de saltos na rede do cliente até o *cloudlet* de forma a permitir o máximo desempenho e relacionar com a viabilidade econômica da existência nuvem intermediária *cloudlet*;

## 7. REFERÊNCIAS

AMAZON. **Amazon Web Services - Cloud Computing Services**, 2016. Disponível em: <https://aws.amazon.com/>. Acessado: 10/02/2016

ARCHER, J. et al. **Security guidance for critical areas of mobile computing** Cloud Security Alliance, , 2012. Disponível em: <https://cloudsecurityalliance.org/download/security-guidance-for-critical-areas-of-mobile-computing/>

**Arena Simulation Software**, , 2016. Disponível em: <https://www.arenasimulation.com/>. Acessado: 10/02/2016

ARMBRUST, M. et al. A view of cloud computing. **Communications of the ACM**, v. 53, n. 4, p. 50–58, 2010.

B'FAR, R. **Mobile computing principles: designing and developing mobile applications with UML and XML**. [s.l.] Cambridge University Press, 2004.

BAHL, P. et al. **Advancing the state of mobile cloud computing** Proceedings of the third ACM workshop on Mobile cloud computing and services. **Anais...ACM**, 2012

BAHWAIRETH, K. et al. Experimental comparison of simulation tools for efficient cloud and mobile cloud computing applications. **EURASIP Journal on Information Security**, v. 2016, n. 1, p. 1–14, 2016.

BERTHOMIEU, B.; DIAZ, M. Modeling and verification of time dependent systems using time Petri nets. **IEEE Transactions on Software Engineering**, v. 17, n. 3, p. 259-273, mar. 1991.

BOLCH, G. et al. **Queueing networks and Markov chains: modeling and performance evaluation with computer science applications**. [s.l.] John Wiley & Sons, 2006.

CHEN, Z. et al. **Early Implementation Experience with Wearable Cognitive Assistance Applications** Proceedings of the 2015 Workshop on Wearable Systems and Applications. **Anais...: WearSys '15**. Florence, Italy: ACM, 2015 Disponível em: <http://doi.acm.org/10.1145/2753509.2753517>

DINH, H. T. et al. A survey of mobile cloud computing: architecture, applications, and approaches. **Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on**, v. 13, n. 18, p. 1587–1611, dez. 2013.

DONGARRA, J. J.; LUSZCZEK, P.; PETITET, A. The LINPACK Benchmark: past, present and future. **Concurrency and Computation: practice and experience**, v. 15, n. 9, p. 803–820, 2003.

EMARKETER. **Smartphone Users Worldwide Will Total 1.75 Billion in 2014**, 2014. Disponível em: <http://www.emarketer.com/Article/Smartphone-Users-Worldwide-Will-Total-175-Billion-2014/1010536>. Acessado: 11/10/2015

EMARKETER. **Mobile Ad Spend to Top 100 Billion Worldwide in 2016, 51% of Digital Market**, 2015. Disponível em: <http://www.emarketer.com/Article/Mobile-Ad-Spend-Top-100-Billion-Worldwide-2016-51-of-Digital-Market/1012299>. Acessado: 11/10/2015

GOOGLE. **The Android mobile OS by Google Inc**, 2016a. Disponível em: <http://www.android.com/>. Acessado: 12/02/2016

GOOGLE. **App Engine by Google**, 2016b. Disponível em: <https://cloud.google.com/appengine/>. Acessado: 12/02/2016

HAMAD, H.; SAAD, M.; ABED, R. Performance Evaluation of RESTful Web Services for Mobile Devices. *Int. Arab J. e-Technol.*, v. 1, n. 3, p. 72–78, 2010.

HAMREN, O. **CloudBench Android App**, 2015. Disponível em: <http://www8.informatik.umu.se/frih0606/about.html>; Acessado: 25/01/2015

HAMRÉN, O. **Mobile phones and cloud computing: A quantitative research paper on mobile phone application offloading by cloud computing utilization**. [s.l: s.n.].

IDC. **Mobile Internet Users to Top 2 Billion Worldwide in 2016, According to IDC**, 2015. Disponível em: <https://www.idc.com/getdoc.jsp?containerId=prUS40855515>. Acessado: 25/06/2016

JAIN, R. **The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling**. [s.l.] John Wiley & Sons, 1991.

KELTON, W. D.; SADOWSKI, R. P.; SADOWSKI, D. A. **Simulation with Arena**. 2. ed. New York, NY, USA: McGraw-Hill, Inc., 2002.

KUMAR, K.; LU, Y.-H. Cloud computing for mobile users: Can offloading computation save energy? *Computer*, v. 43, n. 4, p. 51–56, 2010.

LILJA, D. J. **Measuring computer performance: a practitioner's guide**. [s.l.] Cambridge University Press, 2000.

LINARES-VÁSQUEZ, M. et al. **How developers detect and fix performance bottlenecks in Android apps** Software Maintenance and Evolution (ICSME), 2015 IEEE International Conference on. *Anais...IEEE*, 2015

MELL, P.; GRANCE, T. The NIST definition of cloud computing. 2011.

MENASCE, D. A. et al. **Performance by design: computer capacity planning by example**. [s.l.] Prentice Hall Professional, 2004.

MENASCE, D. A.; ALMEIDA, V. A.; DOWDY, L. W. Capacity Planning and Performance Modeling: From Mainframes to Client-Server Systems. 1994.

MICROSOFT. **Microsoft Azure: Cloud Computing Platform & Services** (M. Azure, Ed.), 2016. Disponível em: <https://azure.microsoft.com/>. Acessado: 07/04/2016

MIETTINEN, A. P.; NURMINEN, J. K. **Energy efficiency of mobile clients in cloud computing** Proceedings of the 2nd USENIX conference on Hot topics in cloud computing. **Anais...**USENIX Association, 2010

MINITAB. **Minitab**, 2016. Disponível em: <http://www.minitab.com>. Acessado: 25/01/2016

MOLLOY, M. K. Performance analysis using stochastic Petri nets. **IEEE Transactions on computers**, v. 100, n. 9, p. 913–917, 1982.

NETEASE. **Emmagee - Test tool for specified Android App**, 2016. Disponível em: <https://github.com/NetEase/Emmagee>

PANDI, K.; CHARAF, H. **Mobile performance metrics for resource management** System Science and Engineering (ICSSE), 2013 International Conference on. **Anais...**IEEE, 2013

QI, H.; GANI, A. **Research on mobile cloud computing: Review, trend and perspectives** Digital Information and Communication Technology and it's Applications (DICTAP), 2012 Second International Conference on. **Anais...**ieee, 2012

RACKSPACE. , 2016. Disponível em: <https://www.rackspace.com/>. Acessado: 12/07/2016

SALESFORCE. , 2016. Disponível em: <https://www.salesforce.com>. Acessado: 12/07/2016

SARGENT, R. G. **Verification and validation of simulation models** Proceedings of the 37th conference on Winter simulation. **Anais...**winter simulation conference, 2005

SATYANARAYANAN, M. **Fundamental challenges in mobile computing** Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing. **Anais...**ACM, 1996

SATYANARAYANAN, M. et al. The case for vm-based cloudlets in mobile computing. **IEEE pervasive Computing**, v. 8, n. 4, p. 14–23, 2009.

SATYANARAYANAN, M. et al. **Cloudlets: at the leading edge of mobile-cloud convergence** Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on. **Anais...**nov. 2014

SEMATECH, N. Engineering statistics handbook. **NIST SEMATECH**, 2006.

SILVA, F. A. et al. Benchmark applications used in mobile cloud computing research: a systematic mapping study. **The Journal of Supercomputing**, v. 72, n. 4, p. 1431–1452, 2016.

SILVA, F. A.; MACIEL, P.; MATOS, R. SmartRank: a smart scheduling tool for mobile cloud computing. **The Journal of Supercomputing**, v. 71, n. 8, p. 2985–3008, 2015.

WEICKER, R. P. An overview of common benchmarks. **Computer**, v. 23, n. 12, p. 65–75, 1990.

ZHANG, Q.; CHENG, L.; BOUTABA, R. Cloud computing: state-of-the-art and research challenges. **Journal of internet services and applications**, v. 1, n. 1, p. 7–18, 2010.